GPC® 883

General Purpose Controller AM188ES

MANUALE TECNICO



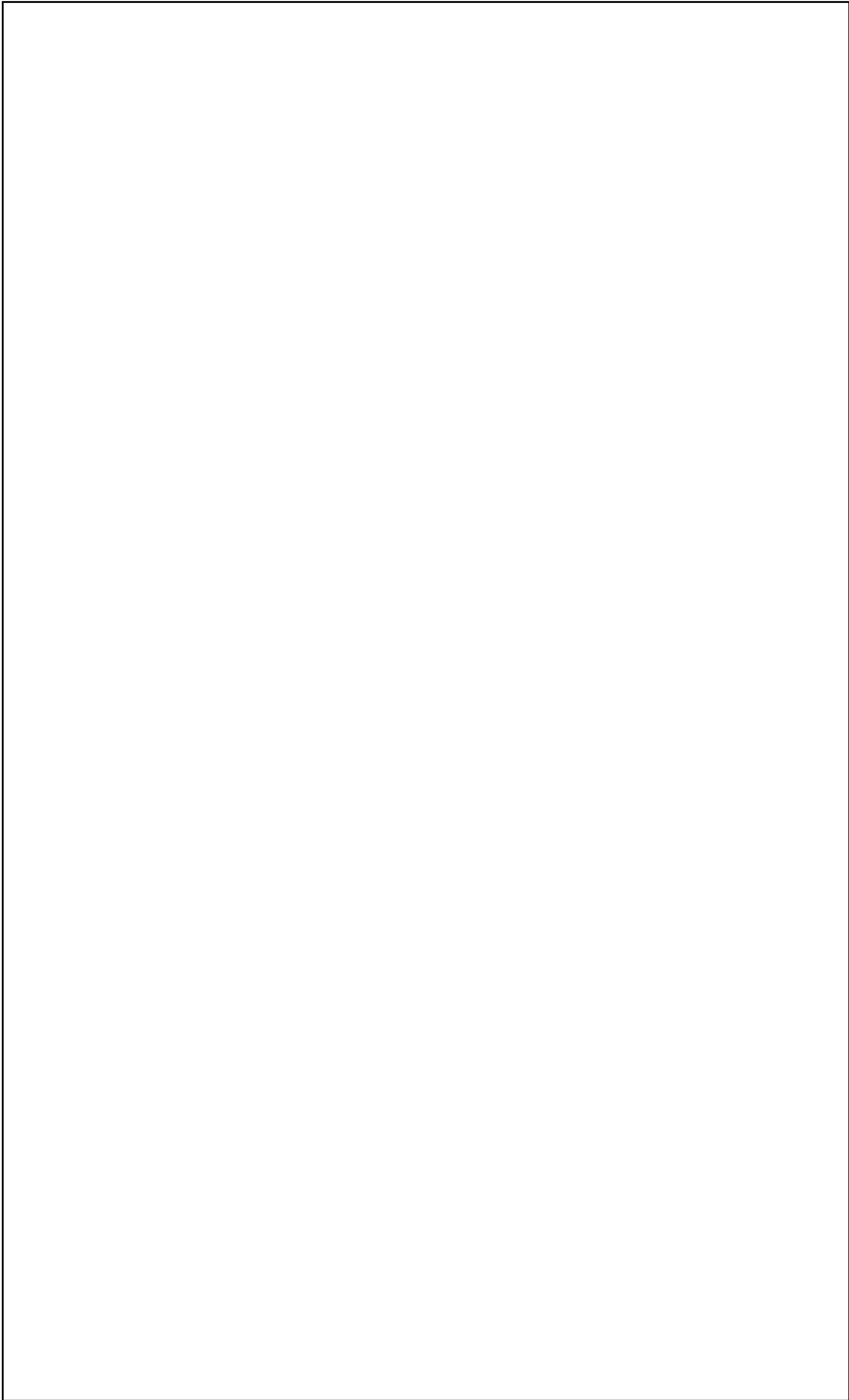
grifo®
ITALIAN TECHNOLOGY

Via dell' Artigiano, 8/6
40016 San Giorgio di Piano
(Bologna) ITALY
E-mail: grifo@grifo.it



http://www.grifo.it http://www.grifo.com
Tel. +39 051 892.052 (r.a.) FAX: +39 051 893.661

GPC® 883 Edizione 3.00 *Rel. 10 Maggio 2002*



GPC® 883

General Purpose Controller AM188ES

MANUALE TECNICO

Modulo Intelligente **Abaco® BLOCK**, della **Serie 3**, nel formato 100x150 mm; contenitore, opzionale, per guide ad Ω tipo **DIN 46277-1** e **DIN 46277-3**. **CPU 80C188ES** da **26** o **40 MHz** con **512K EPROM** o **FLASH** e **512K SRAM**; **EEPROM** seriale fino ad 8K. 1 **dip switch** da 8 vie, leggibile da software e jumper di configurazione per **RUN/DEBUG** mode. **2** linee seriali in **RS 232** di cui una settabile in **RS 422**, **RS 485**, **Current Loop** passivo, con baud rate settabile da software, fino a **115 KBaud**. Linea **CAN** opzionale basata sul controllore PHILIPS **SJA 1000** che supporta i protocolli **Basic CAN**, **CAN 2.0B** e **PeliCAN**, con bit rate fino a **1 MBit/sec**; driver di linea CAN PHILIPS **82C250** disponibile con e senza separazione galvanica. **2** canali **DMA** indipendenti completamente programmabili via software. **34** linee di **I/O TTL**, settabili da software (alcune di queste linee hanno duplice funzione). Sezione opzionale di **A/D** converter con **8** linee da 12 bits e con fondo scala selezionabile via software tra **0÷5**, **0÷10**, **-5÷+5**, **-10÷+10 V** oppure, via hardware, tra **0÷20** e **4÷20 mA**. **3 Timer Couter** da 16 bits in grado di generare impulsi. **2 LED** di stato piu' **LED** e **BUZZER** di segnalazione, gestiti via software. **Real Time Clock** autonomo, con possibilità di generare **INT** a cadenze definibili da software. Circuiteria di **Back Up** per **RAM** e **RTC**, tramite batteria al **LITIO** interna ed esterna con stato di carica acquisibile via software. **Watch Dog** resettabile da software e circuiteria di **power failure** in grado di generare interrupt. Connettore per **Abaco® I/O BUS** da 26 vie; 2 connettori standard di **I/O Abaco®**, da 20 vie; 1 connettore standard di **A/D Abaco®**, da 20 vie; connettore standard **vaschetta D** a **9 vie M** per linea di comunicazione CAN. Unica tensione di alimentazione a **5Vdc**, **270 mA** (26 MHz) o **390 mA** (40 MHz). **Sezione alimentatrice** opzionale switching (**.SW**) o lineare (**.ALIM12**) ad ampio range d'ingresso, in grado di alimentare anche carichi esterni; protezione della logica di bordo dai transienti tramite **TransZorb™**. Vasta disponibilità di software di base e di **tools di sviluppo** che consentono di poter utilizzare la scheda tramite un normale **PC**: **Monitor Debugger Trace**, **GCTR 883**, **HI TECH C 86**, **GET 188**, **PASCAL 188**, **GDOS 188**, **ROM-DOS**, **FLASH WRITER**, **DDS MICRO C 86**, ecc.

grifo®
ITALIAN TECHNOLOGY

Via dell' Artigiano, 8/6
40016 San Giorgio di Piano
(Bologna) ITALY
E-mail: grifo@grifo.it



http://www.grifo.it http://www.grifo.com
Tel. +39 051 892.052 (r.a.) FAX: +39 051 893.661

GPC® 883 Edizione 3.00 *Rel. 10 Maggio 2002*

IMPORTANTE

Tutte le informazioni contenute sul presente manuale sono state accuratamente verificate, ciononostante **grifo**® non si assume nessuna responsabilità per danni, diretti o indiretti, a cose e/o persone derivanti da errori, omissioni o dall'uso del presente manuale, del software o dell' hardware ad esso associato.
grifo® altresi si riserva il diritto di modificare il contenuto e la veste di questo manuale senza alcun preavviso, con l' intento di offrire un prodotto sempre migliore, senza che questo rappresenti un obbligo per **grifo**®.
Per le informazioni specifiche dei componenti utilizzati sui nostri prodotti, l'utente deve fare riferimento agli specifici Data Book delle case costruttrici o delle seconde sorgenti.

LEGENDA SIMBOLI

Nel presente manuale possono comparire i seguenti simboli:



Attenzione: Pericolo generico



Attenzione: Pericolo di alta tensione

Marchi Registrati

INDICE GENERALE

INDICE DELLE FIGURE

INTRODUZIONE

L’uso di questi dispositivi é rivolto - IN VIA ESCLUSIVA - a personale specializzato.

Scopo di questo manuale é la trasmissione delle informazioni necessarie all’uso competente e sicuro dei prodotti. Esse sono il frutto di un’elaborazione continua e sistematica di dati e prove tecniche registrate e validate dal Costruttore, in attuazione alle procedure interne di sicurezza e qualità dell'informazione.

I dati di seguito riportati sono destinati - IN VIA ESCLUSIVA - ad un utenza specializzata, in grado di interagire con i prodotti in condizioni di sicurezza per le persone, per la macchina e per l’ambiente, interpretando un’elementare diagnostica dei guasti e delle condizioni di funzionamento anomale e compiendo semplici operazioni di verifica funzionale, nel pieno rispetto delle norme di sicurezza e salute vigenti.

Le informazioni riguardanti installazione, montaggio, smontaggio, manutenzione, aggiustaggio, riparazione ed installazione di eventuali accessori, dispositivi ed attrezzature, sono destinate - e quindi eseguibili - sempre ed in via esclusiva da personale specializzato avvertito ed istruito, o direttamente dall’ASSISTENZA TECNICA AUTORIZZATA, nel pieno rispetto delle raccomandazioni trasmesse dal costruttore e delle norme di sicurezza e salute vigenti.

I dispositivi non possono essere utilizzati all'aperto. Si deve sempre provvedere ad inserire i moduli all'interno di un contenitore a norme di sicurezza che rispetti le vigenti normative. La protezione di questo contenitore non si deve limitare ai soli agenti atmosferici, bensì anche a quelli meccanici, elettrici, magnetici, ecc.

Per un corretto rapporto coi prodotti, é necessario garantire leggibilità e conservazione del manuale, anche per futuri riferimenti. In caso di deterioramento o più semplicemente per ragioni di approfondimento tecnico ed operativo, consultare direttamente l’Assistenza Tecnica autorizzata.

Al fine di non incontrare problemi nell’uso di tali dispositivi, é conveniente che l’utente - PRIMA DI COMINCIARE AD OPERARE - legga con attenzione tutte le informazioni contenute in questo manuale. In una seconda fase, per rintracciare più facilmente le informazioni necessarie, si può fare riferimento all’indice generale e all’indice analitico, posti rispettivamente all’inizio ed alla fine del manuale.

VERSIONE SCHEDA

Il presente manuale è riferito alla scheda **GPC® 883** versione **110801** e successive. La validità delle informazioni riportate è quindi subordinata al numero di versione della scheda in uso e l’utente deve quindi sempre verificare la giusta corrispondenza tra le due indicazioni. Sulla scheda il numero di versione è riportato in più punti sia a livello di serigrafia che di stampato (ad esempio sotto alla batteria BT1 sia sul lato componenti che sul lato stagnature).

INFORMAZIONI GENERALI

La scheda **GPC® 883**, che fa parte della **Serie 3**, e' un potentissimo modulo di controllo, in grado di funzionare autonomamente come periferica intelligente o remotata in una più vasta rete di telecontrollo e/o di acquisizione. La **GPC® 883** può essere fornita di un supporto in plastica provvisto degli attacchi per le guide ad **Omega** tipo **DIN 46277-1** e **DIN 46277-3**. In questo modo non é necessario l'uso di un **Rack**, ma la scheda può essere montata, in modo più economico e veloce, direttamente nel quadro elettrico. Inoltre la scheda puo' essere montata nella stessa guida in plastica che contiene le schede periferiche di espansione formando in questo modo un unico elemento **BLOCK**.
La **GPC® 883** e' basata sulla potente e nota **CPU 80C188ES AMD**, codice compatibile con qualsiasi **PC**, ed ha a bordo scheda notevoli risorse hardware. Particolarmente interessanti sono la disponibilità di 8 linee di **A/D Converter** da 12 bits, le numerose linee di I/O a livello TTL e la linea di comunicazione **CAN**.
La estrema modularita' e la notevole completezza di risorse hardware della scheda **GPC® 883** le consentono di poter affrontare applicazioni anche di notevole complessita' con estrema disinvoltura. La notevole facilita' di impiego e' determinata anche dalla ricca serie di tools di sviluppo software basati su linguaggi ad alto livello che consentono di poter lavorare al meglio utilizzando unicamente un normale PC. Tra questi tools si ricordano l'ambiente integrato composto dal **GET 188** in abbinamento ad un efficiente **compilatore romato PASCAL** ed un tools che consente di **romare**, quanto sviluppato con i normali **compilatori C** della **Borland** Grande attenzione e' stata riservata alla messa a punto dell'applicativo, rendendo disponibili dei programmi che consentono di effettuare il **debugger simbolico remoto** direttamente sulla scheda e che programmano direttamente la **FLASH** di bordo con il programma utente.
La **GPC® 883** é dotata di una serie di connettori normalizzati, standard **Abaco®**, che le consentono di utilizzare immediatamente la numerosa serie di moduli **BLOCK** di I/O oppure permettono il collegamento diretto di interfaccie operatori locali (**KDx x24**, **QTP xxP**) ,consentendo così una notevole riduzione dei costi. La presenza del connettore **Abaco® I/O BUS** consente inoltre di poter pilotare direttamente le schede di I/O tipo **ZBR xxx**, **ZBT xxx**, e tramite **ABB 03**, **ABB 05**, ecc. e' possibile gestire tutte le numerose schede periferiche disponibili sul **BUS Abaco®**.

- Modulo Intelligente **Abaco® BLOCK**, della **Serie 3**, nel formato 100x150 mm.
- Contenitore, opzionale, per guide ad Ω tipo **DIN 46277-1** e **DIN 46277-3**.
- **CPU 80C188ES** da **26** o **40 MHz** con **512K EPROM** o **FLASH** e **512K SRAM**.
- **EEPROM** seriale fino ad 8K.
- 1 **dip switch** da 8 vie, leggibile da software e jumper di configurazione per **RUN/DEBUG** mode.
- **2** linee seriali in **RS 232** di cui una settabile in **RS 422**, **RS 485**, **Current Loop** passivo, con baud rate settabile da software, fino a **115 KBaud**.
- Linea **CAN** opzionale basata sul controllore PHILIPS **SJA 1000** che supporta i protocolli **Basic CAN**, **CAN 2.0B** e **PeliCAN**, con bit rate fino a **1 MBit/sec**.
- Driver di linea CAN PHILIPS **82C250** disponibile con e senza separazione galvanica.
- **2** canali **DMA** indipendenti completamente programmabili via software.
- **34** linee di **I/O TTL**, settabili da software (alcune di queste linee hanno duplice funzione).
- Sezione opzionale di **A/D** converter con **8** linee da 12 bits e con fondo scala selezionabile via software tra **0÷5**, **0÷10**, **-5÷+5**, **-10÷+10 V** oppure,via hardware, tra **0÷20** e **4÷20 mA**.
- **3 Timer Couter** da 16 bits in grado di generare impulsi.
- **2 LED** di stato piu' **LED** e **BUZZER** di segnalazione, gestiti via software.

- **Real Time Clock** autonomo, con possibilità di generare **INT** a cadenze definibili da software.
- Circuiteria di **Back Up** per **RAM** e **RTC**, tramite batteria al **LITIO** interna ed esterna con stato di carica acquisibile via software.
- **Watch Dog** resettabile da software e circuiteria di **power failure** in grado di generare interrupt.
- Connettore per **Abaco® I/O BUS** da 26 vie.
- 2 connettori standard di **I/O Abaco®**, da 20 vie
- 1 connettore standard di **A/D Abaco®**, da 20 vie
- Connettore standard **vaschetta D** a **9 vie M** per linea di comunicazione CAN.
- Unica tensione di alimentazione a **5Vdc**, **270 mA** (26 MHz) o **390 mA** (40 MHz).
- **Sezione alimentatrice** opzionale switching (**.SW**) o lineare (**.ALIM12**) ad ampio range d'ingresso, in grado di alimentare anche carichi esterni.
- Protezione della logica di bordo dai transienti tramite **TransZorb™**.
- Vasta disponibilità di software di base e di **tools di sviluppo** che consentono di poter utilizzare la scheda tramite un normale **PC**: **Monitor Debugger Trace**, **GCTR 883**, **HI TECH C 86**, **GET 188**, **PASCAL 188**, **GDOS 188**, **ROM-DOS**, **FLASH WRITER**, **DDS MICRO C 86**, ecc.

Viene di seguito riportata una descrizione dei blocchi funzionali della scheda, con indicate le operazioni effettuate da ciascuno di essi. Per una più facile individuazione di tali blocchi e per una verifica delle loro connessioni, fare riferimento alla figura 1.

PROCESSORE DI BORDO

La scheda **GPC® 883** é predisposta per accettare il processore **Am188ES** prodotto dalla **AMD**. Tale processore a 16 bit é codice compatibile con la difusissima famiglia Intel 86 ed é quindi caratterizzato da: un esteso set di istruzioni, un’alta velocità di esecuzione, un'efficiente gestione degli interrupts, una vasta scelta di modalità d'indirizzamento, una efficacissima manipolazione dei dati, ecc. Di fondamentale importanza é la presenza delle seguenti periferiche interne al microprocessore:

- 2 Timer Counter ed 1 Timer a 16 bit, con possibilità di generare forme d'onda (TCU);
- 2 linee seriali asincrone complete di segnali di handshake (ASP);
- 2 canali di DMA per trasferimenti dati ad alta velocità (DMAC);
- sezione di controllo interrupt (ICU);
- sezione di power management (CPMU);
- sezione di generazione segnali di controllo periferiche e memorie (CSU);
- gestore automatico del rinfresco di memorie dinamiche (RCU);
- 32 linee di I/O definibili via software (PIO);
- 1 sezione di acquisizione segnali PWM (PWD);

Da ricordare che il processore accede a tutte le risorse della scheda tramite un BUS dati a 8 bit pur effettuando operazioni interne a 16 bit; questa trasformazione é gestita automaticamente dall'**Am188ES** senza alcun intervento dell'utente.
Per maggiori informazioni sul componente si faccia riferimento all’apposita documentazione della casa costruttrice, oppure all’appendice B di questo manuale.

CLOCK

Sulla **GPC® 883** sono presenti tre circuiterie di clock:
- La prima é basata su un quarzo che provvede a generare la frequenza di clock per la CPU da cui vengono ricavate anche le frequenze necessarie per le altre sezioni della scheda (Timer, Seriali, DMA ecc.). Il valore standard di clock é di **26,045 MHz**, ma in caso di applicazioni particolarmente veloci, la frequenza di clock può essere aumentata fino a **40,665 MHz** intervenendo via hardware sull'apposita circuiteria. La selezione della frequenza di lavoro non può essere effettuata dall'utente e deve quindi essere specificata in fase di ordine, tramite l'apposita opzione **.40M**. Si ricorda inoltre che la frequenza di clock della CPU normalmente coincide con quella del quarzo montato a bordo scheda e che si può intervenire via software sull'apposita sezione di power management per ridurne i valori e quindi i consumi.
- La seconda circuiteria é basata su un oscillatore che genera le frequenze necessarie al controllore CAN ed al convertitore A/D, rispettivamente di **16 MHz** e **2 MHz**. Tali valori sono il risultato di un'approfondita progettazione e sperimentazione e non possono essere modificati.
- Infine la terza circuiteria provvede a fornire le giuste temporizzazioni al real time clock di bordo ed é basata su un quarzo da **32,768 KHz**, integrato nello stesso componente.

ABACO® I/O BUS

Una delle caratteristiche di fondamentale importanza della **GPC® 883** è quella di disporre del cosiddetto **ABACO®** I/O BUS: ovvero un connettore normalizzato **ABACO®** con cui è possibile collegare la scheda ad una serie di moduli esterni intelligenti e non. Tra questi si trovano moduli per acquisizione di segnali analogici (A/D), per la generazione di segnali analogici (D/A), schede con contatori e temporizzatori, schede per gestione di linee di I/O logico, ecc. e ne possono essere realizzati anche su specifiche richieste dell'utente.
Utilizzando mother board come l'**ABB 03** o l'**ABB 05** é inoltre possibile gestire tutte le schede periferiche in formato Europa con interfaccia per BUS **ABACO®**. Tale caratteristica rende la scheda espandibile con un ottimo rapporto prezzo/prestazioni e quindi adatta a risolvere molti dei problemi dell'automazione industriale.
Da ricordare che tutte le linee dell'**ABACO®** I/O BUS sono bufferate in modo da garantire una maggiore protezione contro i disturbi esterni ed il comando di un maggior numero di schede periferiche di espansione, senza problemi di fan-out.
Il capitolo "SCHEDE ESTERNE" mostra una breve descrizione di alcune di queste schede di espansione.

LOGICA DI CONTROLLO

Il mappaggio e la gestione di alcune delle periferiche presenti sulla scheda e dei dispositivi di memoria, é affidata ad un'opportuna logica di controllo che si occupa di allocare tali dispositivi nello spazio d'indirizzamento della CPU.
Per maggiori informazioni fare riferimento ai paragrafi “MAPPAGGIO I/O” e "MAPPAGGIO MEMORIE".

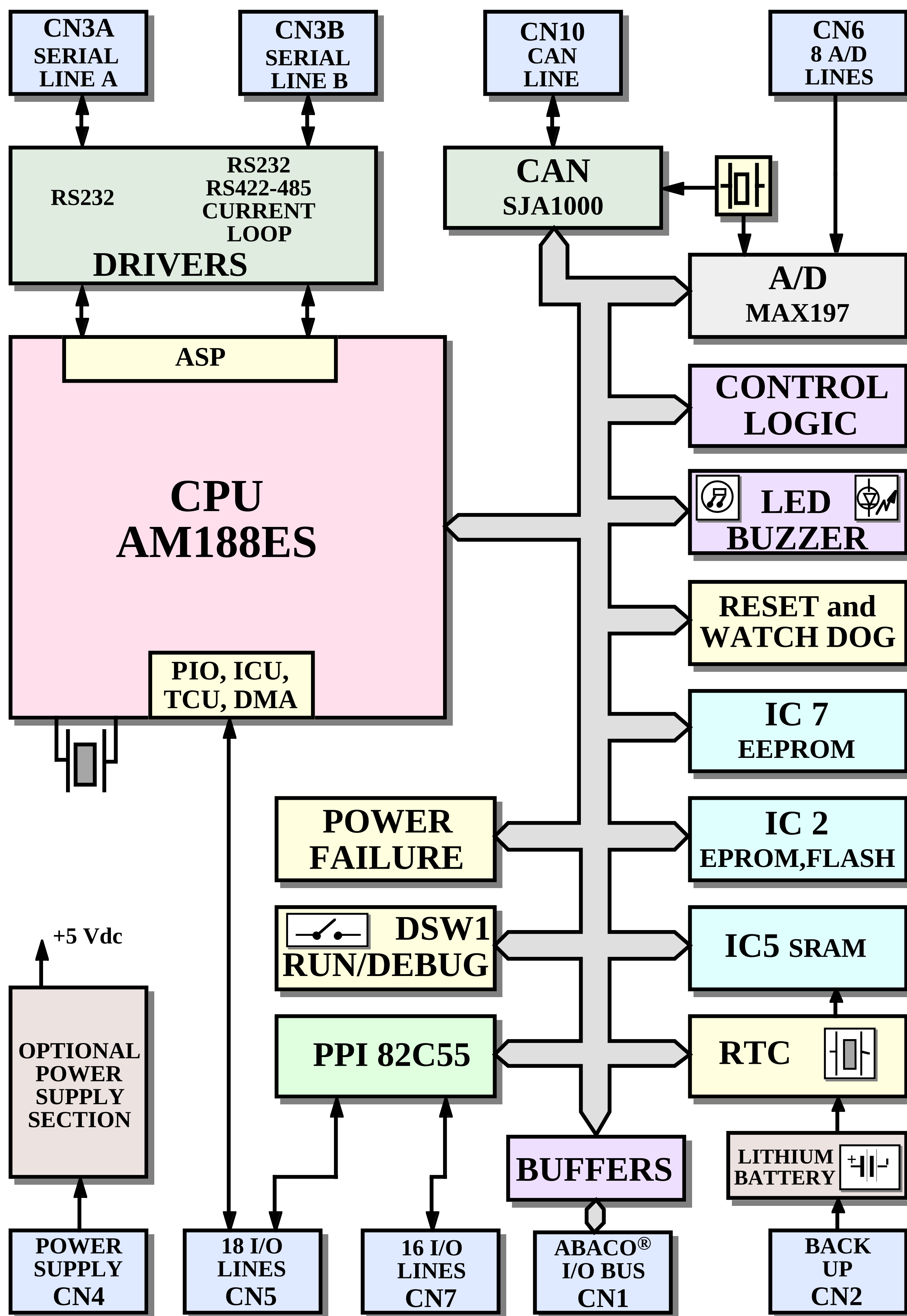


Figura 1: Schema a blocchi

MEMORIE

E’ possibile dotare la scheda di un massimo di 1032K di memoria variamente suddivisi con un massimo di 512KByte di EPROM o FLASH EPROM, 512KByte di RAM statica ed infine 8KByte di EEPROM seriale. La scelta della configurazione delle memorie presenti sulla scheda può avvenire in relazione all’applicazione da risolvere e quindi alle esigenze dell’utente. Da questo punto di vista si ricorda che la scheda viene normalmente fornita con 128K Byte di SRAM più 512 Byte di EEPROM seriale e che tutte le rimanenti configurazioni di memoria devono essere quindi opportunamente specificate in fase di ordine della scheda.
Sfruttando la circuiteria di back up di bordo più l'eventuale batteria tampone esterna, si ha la possibilità di mantenere i dati anche in assenza di alimentazione. Questa caratteristica fornisce alla scheda la possibilità di ricordare in ogni condizione, una serie di parametri come ad esempio la configurazione o lo stato del sistema, anche per lunghi periodi di inattività senza dover ricorrere a costosi gruppi di continuità esterni.
Il mappaggio delle risorse di memoria avviene tramite una opportuna circuiteria di bordo (Chip Select Unit), che provvede ad allocare i dispositivi all’interno dello spazio d’indirizzamento del microprocessore. Questa circuiteria é inoltre in grado di gestire automaticamente tutte le modalità d'indirizzamento richieste dai pacchetti software per la **GPC® 883** e quest'ultimi provvedono a programmarla autonomamente, senza richiedere l'intervento dell'utente.
Per maggiori informazioni fare riferimento al capitolo "MAPPAGGI ED INDIRIZZAMENTI" e "DESCRIZIONE SOFTWARE DELLE PERIFERICHE DI BORDO". Per una descrizione più approfondita sui dispositivi di memoria, sugli zoccoli da utilizzare e sullo strippaggio della scheda, fare riferimento al paragrafo "SELEZIONE MEMORIE".

REAL TIME CLOCK

La **GPC® 883** dispone di un completo Real Time Clock in grado di gestire ore, minuti, secondi, giorno del mese, mese, anno e giorno della settimana in modo completamente autonomo. Il componente é alimentato dalla circuiteria di back up in modo da garantire la validità dei dati in ogni condizione operativa ed é completamente gestito via software, tramite la programmazione di 16 registri situati nello spazio di I/O della CPU dall'apposita logica di controllo (CSU). La sezione di RTC può inoltre generare interrupt in corrispondenza di intervalli di tempo programmabili via software, in modo da poter periodicamente distogliere la CPU dalle normali operazioni oppure periodicamente risvegliarla dagli stati di basso consumo.

CIRCUITERIE DI SICUREZZA: WATCH DOG E POWER FAILURE

La scheda **GPC® 883** é provvista di una circuiteria di watch dog che, se utilizzata, consente di uscire da stati di loop infinito o da condizioni anomale non previste dal programma applicativo. Tale circuiteria é composta da una sezione astabile con un tempo d’intervento di circa 1400 msec. Tutta la gestione avviene via software (tramite l’accesso ad un opportuno registro situato nello spazio d’indirizzamento della CPU) e conferisce al sistema basato sulla scheda, una sicurezza estrema in qualsiasi condizione operativa.
Una circuiteria di power failure può invece informare il programma applicativo sviluppato dall'utente dell'imminente mancanza della tensione di alimentazione; lo stesso programma potrà quindi reagire tempestivamente provvedendo ad effettuare le necessarie operazioni di arresto, salvataggio, ecc.

LINEE DI I/O DIGITALI

La scheda dispone di due controllori di I/O digitale che sono utilizzati per comandare alcune risorse di bordo (LED di attività, stato batteria, EEPROM seriale, buzzer, ecc.) e per gestire 34 linee di I/O digitale TTL, a disposizione dell'utente, di cui:
- 18 con direzionalità settabile per ogni linea, collegate alla sezione PIO della CPU;
- 24 con direzionalità settabile a gruppi di 8 linee, collegate al PPI 82C55.

Tali linee sono collegate direttamente a due connettori a 20 vie con pin out standard **I/O ABACO®** ed hanno quindi la possibilità di essere direttamente collegate a numerose schede d'interfaccia.
Via software é definibile la funzionalità di queste linee, con possibilità di associarle anche alle periferiche della scheda (DMA, Timer Counter, Interrupt, ecc.), tramite una semplice programmazione di 10 registri situati nello spazio di I/O della CPU.
Per maggiori informazioni fare riferimento ai paragrafi "CONNESSIONI", "INTERFACCE PER I/O DIGITALI", "PIN MULTIFUNZIONE" e "DESCRIZIONE SOFTWARE DELLE PERIFERICHE DI BORDO".

CONFIGURAZIONE SCHEDA

Allo scopo di rendere configurabile la scheda ed in particolare il programma applicativo sviluppato, é stato previsto un dip switch ad 8 vie ed un jumper di configurazione. La possibilità di acquisire via software il loro stato, fornisce all'utente la possibilità di gestire diverse condizioni tramite un unico programma, senza dover rinunciare ad altre linee d'ingresso (le applicazioni caratteristiche sono: selezione della lingua di rappresentazione, definizione parametri del programma, selezione delle modalità operative, ecc).
Alcuni pacchettisoftware sviluppati per la **GPC® 883** usano il jumper J1 per selezionare la modalità operativa RUN o DEBUG, come descritto negli appositi manuali d'uso degli stessi pacchetti.
In aggiunta la scheda ha un LED di attività ed un buzzer, gestibili via software, che possono essere usati per segnalare visivamente o acusticamente la configurazione attuale della scheda, come descritto negli appositi paragrafi.
Tutte le risorse di configurazione descritte sono completamente gestite via software, tramite la programmazione di appositi registri allocati nello spazio di I/O dalla logica di controllo.

A/D CONVERTER

La sezione di A/D Converter della **GPC® 883** é basata su un convertitore DAS (Data Acquisition System) di precisione che sfrutta il principio delle approssimazioni successive con circuiteria di track-hold interna. Le caratteristiche principali di questa sezione sono: risoluzione di 12 bit in modalità unipolare o di 11 bit più segno in modalità bipolare; ingressi multi range configurabili via software nelle scale ±5 V, ±10 V, 0÷5 V, e 0÷10 V e via hardware nelle scale 0÷20 mA, 4÷20 mA; tempo di conversione su singolo canale di 6 μsec; 100 Ksps di sample rate su ogni canale; semplice gestione software; generazione interrupt di fine conversione. La sezione A/D converter é completamente gestita via software, tramite la programmazione di tre registri allocati nello spazio di I/O dalla logica di controllo.
La sezione A/D é opzionale, ovvero non presente se non specificata in fase di ordine della scheda; può eventualmente essere aggiunta in un secondo tempo, ma solo dal personale specializzato della **grifo®**. Il codice dell'opzione A/D converter da specificare in fase di ordine è **.AD**.

COMUNICAZIONE SERIALE

La comunicazione seriale é completamente settabile via software, in modo completamente autonomo, per quanto riguarda il protocollo fisico (baud rate, stop bit, bit per carattere, parità e gestione handshake) su entrambe le linee di comunicazione asincrone della scheda. Tali settaggi avvengono tramite la programmazione dei registri interni del microprocessore relativi alla sezione ASP e PIO come descritto nella documentazione tecnica della casa costruttrice o nell'appendice B di questo manuale.
Dal punto di vista hardware, tramite una serie di comodi jumpers e driver da installare, é invece possibile selezionare il protocollo elettrico di comunicazione. In particolare una linea (A) é sempre bufferata in RS 232, mentre la rimanente linea (B) può essere bufferata in **RS 232**, **current loop** o **RS 422**, **RS 485**; in quest'ultimi casi é definibile anche l'attivazione e/o la direzionalità della linea di comunicazione. Si ricorda che la scheda viene normalmente fornita con entrambe le linee seriali bufferate in RS 232 e che tutte le rimanenti configurazioni devono essere quindi opportunamente specificate in fase di ordine della scheda.
Per ulteriori informazioni in merito alla comunicazione seriale fare riferimento ai paragrafi "CONNESSIONI" e "SELEZIONE COMUNICAZIONE SERIALE".

LINEA CAN

Questa sezione é basata sul potente controllore **SJA 1000** della PHILIPS e si preoccupa della gestione software del protocollo CAN in tutte le sue modalità ed aspetti. Le caratteristiche fondamentali di questa sezione sono:

- supporto protocollo BasicCAN;
- supporto del protocollo PeliCAN 2.0B;
- gestione identificatori da 11 e 29 bits;
- buffer di trasmissione da 13 bytes;
- buffer di ricezione da 64 bytes;
- baud rate programmabile fino ad 1M Bit/sec;
- eliminazione del comparatore di ricezione;
- filtri di accettazione messaggi configurabili;
- driver di uscita programmabile;
- frequenza di lavoro16M Hz.

Dal punto di vista elettrico la scheda é dotata dell'apposito driver di linea **82C250** della PHILIPS, galvanicamente isolato. Questo componente si preoccupa di soddisfare tutte le specifiche di collegamento con il campo, definite nel protocollo CAN senza richiedere alcun intervento software. Inoltre la linea CAN di bordo é galvanicamente isolata dal resto della scheda, in modo da garantire l'immunità agli eventuali disturbi del campo; questa caratteristica é di fondamentale importanza soprattutto nel caso di collegamento con sistemi remoti a diversi potenziali oppure di collegamenti con cavi che attraversano ambienti elettricamente rumorosi. Un apposito DC/DC converter si preoccupa di generare le tensioni galvanicamente isolate richieste dal driver di linea, mentre l'interfacciamento con le linee di comunicazionedel controllore CAN sono effettuati tramite appositi optoisolatori per alte frequenze.
Nel caso di particolari esigenze dell'utente é possibile avere la **GPC® 883** con la linea CAN non isolata; per ulteriori informazioni contattare direttamente la **grifo®**.

Il collegamento con il campo della linea CAN é effettuato tramite un connettore a vaschetta D a 9 vie che facilita il cablaggio e garantisce una buona trasmissione del segnale.
Si ricorda che la sezione CAN é opzionale, ovvero non presente se non specificata in fase di ordine della scheda: il codice di tale opzione è **.CAN**.
Dal punto di vista software la linea CAN é completamente configurabile tramite la programmazione di 64 registri allocati nello spazio di I/O dalla logica di controllo ed é in grado di generare interrupt in corrispondenza di numerose condizioni di stato.
Per ogni chiarimento necessario l'utente può fare riferimento all'apposita documentazione della casa costruttrice, oppure all'appendice B di questo manuale.

ALIMENTAZIONE DI BORDO

Una delle caratteristiche peculiari della **GPC® 883** é quella di poter essere dotata di una sezione alimentatrice a bordo scheda che provvede a generare l'unica tensione di alimentazione necessaria di +5 Vdc. In caso di assenza della sezione alimentatrice quest'ultima é l'unica tensione richiesta dalla scheda, viceversa sono disponibili due diverse sezioni alimentatrici: quella lineare che richiede una tensione 6÷12 Vac oppure quella switching che necessita di una tensione 8÷24 Vac (per maggiori informazioni vedere apposito paragrafo "ALIMENTAZIONE"). In ogni caso la tensione di alimentazione può essere fornita tramite appositi connettori standardizzati di facile ed immediata installazione.
Sulla scheda sono state adottate tutte le scelte circuitali e componentistiche che tendono a ridurre i consumi, compresa la possibilità di far lavorare il microprocessore in power save mode, ed a ridurre la sensibilità ai disturbi.
Normalmente la scheda viene fornita senza sezione alimentatrice e l'eventuale opzione deve essere quindi aggiunta dal personale specializzato della **grifo®**, a seguito dell'apposita indicazione, in fase di ordine della scheda.

SPECIFICHE TECNICHE

CARATTERISTICHE GENERALI

Risorse della scheda:	10 input/output digitali TTL (PIO)	
	24 input/output digitali TTL (PPI 82C55)	
	2 timer counter a 16 bit (TCU)	
	1 timer a 16 bit (TCU)	
	2 canali trasferimento dati (DMAC)	
	1 linea seriale RS 232 (ASP 1=A)	
	1 linea seriale RS 232, RS 422, RS 485, current loop (ASP 0=B)	
	1 linea CAN	
	8 linee di A/D converter	
	1 watch dog hardware astabile	
	1 LED gestibile via software	
	1 tasto locale di reset	
	1 real time clock	
	1 buzzer	
	1 jumper di configurazione	
	1 dip switch da 8 vie	
	1 circuiteria di power failure	
	1 circuiteria di back up	
	1 sezione alimentatrice	
	1 interfaccia **ABACO®** I/O BUS	
Memoria indirizzabile:	IC 5:	EPROM da 128K x 8 a 512K x 8
		FLASH EPROM da 128K x 8 a 512K x 8
	IC 4:	SRAM da 128K x 8 a 512K x 8
	IC 22:	EEPROM seriale da 256 byte a 8K byte
Tempo d'accesso memorie:	70 nsec	
CPU di bordo:	AMD Am188ES	
Frequenza clock CPU:	26,045 o 40,665 MHz	
Frequenza max contatori TCU:	Frequenza clock CPU / 4	
Frequenza clock A/D:	2 MHz	
Frequenza taglio ingressi A/D:	1 MHz	
Frequenza clock CAN:	16 MHz	
Bit rate massimo CAN:	1 Mbit	
Risoluzione A/D:	12 bit	
Tempo conversione A/D:	9 μsec	
Errore complessivo A/D:	±2 punti	
Tempo intervento watch dog:	da 940 msec a 2060 msec (tipico 1420 msec)	

CARATTERISTICHE FISICHE

Dimensioni (L x A x P):	100 x 150 x 32 mm	(senza contenitore)
	110 x 60 x 65 mm	(con contenitore per guide DIN)
Peso:	200 g	(senza contenitore)
	320 g	(con contenitore per guide DIN)

Connettori:	CN1: 26 vie scatolino verticale M		
	CN2: 2 vie scatolino verticale M		
	CN3A: Plug a 6 vie 90 gradi F		
	CN3B: Plug a 6 vie 90 gradi F		
	CN4: 2 vie rapida estrazione M o morsettiera		
	CN5: 20 vie scatolino verticale M		
	CN6: 20 vie scatolino verticale M		
	CN7: 20 vie scatolino verticale M		
	CN10: 9 vie vaschetta D 90 gradi M		
Range di temperatura:	da 0 a 50 gradi Centigradi		
Umidità relativa:	20% fino a 90%	(senza condensa)	

CARATTERISTICHE ELETTRICHE

Tensione di alimentazione:	5 Vdc ±5%	(senza sezione alimentatrice)	
	6÷12 Vac *	(sezione alimentatrice lineare)	
	8÷24 Vac *	(sezione alimentatrice switching)	
Corrente assorbita sui +5 Vdc:	220 mA	conf. base	26,045 MHz
	270 mA	conf. massima	26,045 MHz
	125 mA	power down	26,045 MHz
	340 mA	conf. base	40,665 MHz
	390 mA	conf. massima	40,665 MHz
	165 mA	power down	40,665 MHz
Corrente fornita sui +5 Vdc	1000 mA *		
Corrente disponibile sui +5 Vdc	1000 mA - Corrente assorbita		
Batteria di bordo di back up:	3,0 Vdc; 1/2 AA		
Batteria esterna di back up:	3,6÷5 Vdc		
Corrente di back up:	3,0 μA	(batteria di bordo)	
	4,2 μA	(batteria esterna da 3,6 Vdc)	
Ingressi analogici:	0÷5; 0÷10; ±5; ±10 Vdc	(settabili via software)	
Ingressi analogici in corrente:	0÷20; 4÷20 mA	(con modulo di conversione)	
Impedenza ingressi analogici:	21 KΩ	(modalità unipolare)	
	16 KΩ	(modalità bipolare)	
Rete terminazione RS 422-485:	Resistenza terminazione linea=	120 Ω	
	Resistenza di pull up sul positivo=	3,3 KΩ	
	Resistenza di pull down sul negativo=	3,3 KΩ	
Impedenza di linea CAN:	60 Ω		
Rete terminazione CAN:	Resistenza da 120 Ω, disinseribile		
Soglia d’intervento power failure:	52 mV prima dell'intervento del reset		

* I dati riportati sono riferiti ad un lavoro a temperatura ambiente di 20 gradi centigradi (per ulteriori informazioni fare riferimento al paragrafo "ALIMENTAZIONE").

INSTALLAZIONE

In questo capitolo saranno illustrate tutte le operazioni da effettuare per il corretto utilizzo della scheda. A questo scopo viene riportata l'ubicazione e la funzione degli strip, dei connettori, dei trimmers, dei LED, ecc. presenti sulla **GPC® 883**.

CONNESSIONI

Il modulo **GPC® 883** è provvisto di 9 connettori con cui vengono effettuati tutti i collegamenti con il campo e con le altre schede del sistema di controllo da realizzare. Di seguito viene riportato il loro pin out ed il significato dei segnali collegati; per una facile individuazione di tali connettori, si faccia riferimento alla figura 24, mentre per ulteriori informazioni a riguardo del tipo di connessioni, fare riferimento alle figure successive che illustrano il tipo di collegamento effettuato a bordo scheda.

CN4 - CONNETTORE ALIMENTAZIONE CON SEZIONE ALIMENTATRICE

CN4 é un connettore a morsettiera per rapida estrazione, composto da 2 contatti.
Tramite CN4 deve essere fornita la tensione di alimentazione della scheda quando quest'ultima é provvista di una delle due possibili sezioni alimentatrici. Utilizzando la scheda senza la sezione alimentatrice, la tensione +5 Vdc deve essere fornita tramite il pin 26 (+Vdc) e il pin 25 (GND) di CN1.

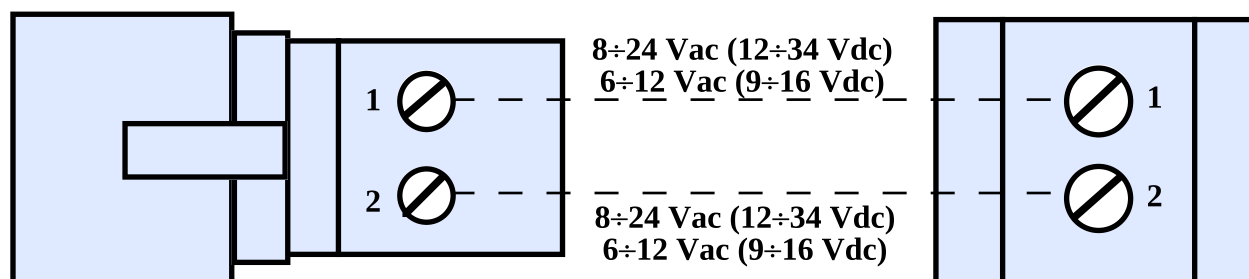


FIGURA 2: CN4 - CONNETTORE ALIMENTAZIONE CON SEZIONE ALIMENTATRICE

Legenda:

8÷24 Vac (12÷34 Vdc) = I - Linee per l'alimentazione con sezione switching
6÷12 Vac (9÷16 Vdc) = I - Linee per l'alimentazione con sezione lineare

Ulteriori informazioni sulle modalità ed opzioni di alimentazione sono riportate nell'apposito paragrafo "ALIMENTAZIONE".

CN1 - CONNETTORE PER ABACO® I/O BUS

CN1 è un connettore a scatolino verticale con passo 2.54 mm a 26 piedini.
Tramite CN1 si effettua la connessione tra la scheda e la serie di moduli esterni di espansione. Tale collegamento è effettuato tramite l'**ABACO®** I/O BUS di cui questo connettore riporta tutti i segnali a livello TTL, opportunamente bufferati.

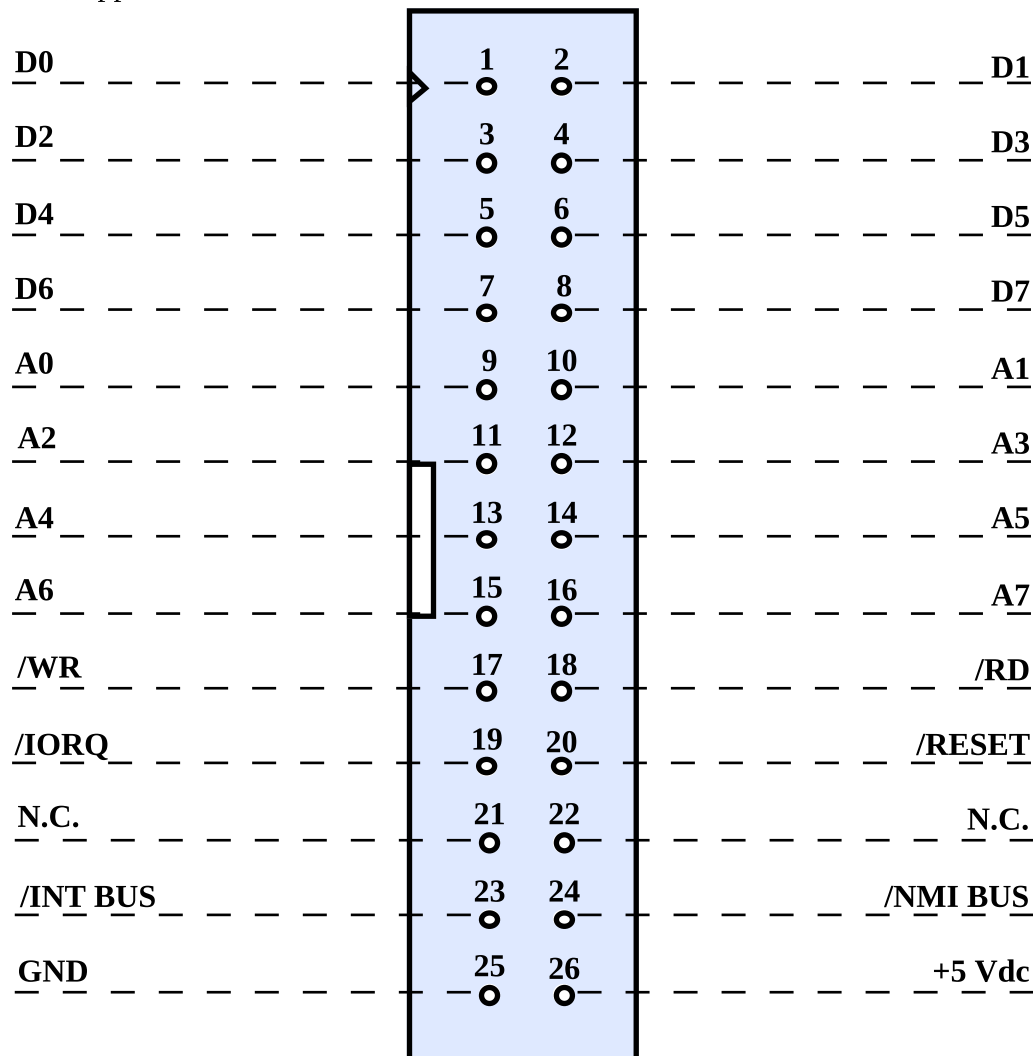


FIGURA 3: CN1 - CONNETTORE PER ABACO® I/O BUS

Legenda:

A0÷A7	= O	- Address BUS: BUS degli indirizzi.
D0÷D7	= I/O	- Data BUS: BUS dei dati.
/INT BUS	= I	- Interrupt request: richiesta d'interrupt
/NMI BUS	= I	- Non Mascable Interrupt: richiesta d'interrupt non mascherabile.
/IORQ	= O	- Input Output Request: richiesta operazione Input Output su I/O BUS.
/RD	= O	- Read cycle status: richiesta di lettura.
/WR	= O	- Write cycle status: richiesta di scrittura.
/RESET	= O	- Reset: azzeramento.
+5 Vdc	= I	- Linea di alimentazione a +5 Vdc.
GND	=	- Linea di massa.
N.C.	=	- Non collegato.

CN3A - CONNETTORE PER LINEA SERIALE A

CN3A é un connettore femmina, a 90 gradi, del tipo plug a 6 vie.
Sul connettore sono disponibili i segnali per la comunicazione della linea seriale A, in RS 232, che é fisicamente collegata alla sezione ASP1 della CPU. La disposizione dei segnali, riportata di seguito, é stata studiata in modo da ridurre al minimo le interferenze ed in modo da facilitare la connessione con il campo, mentre i segnali rispettano le normative definite dal CCITT relative allo standard RS 232.

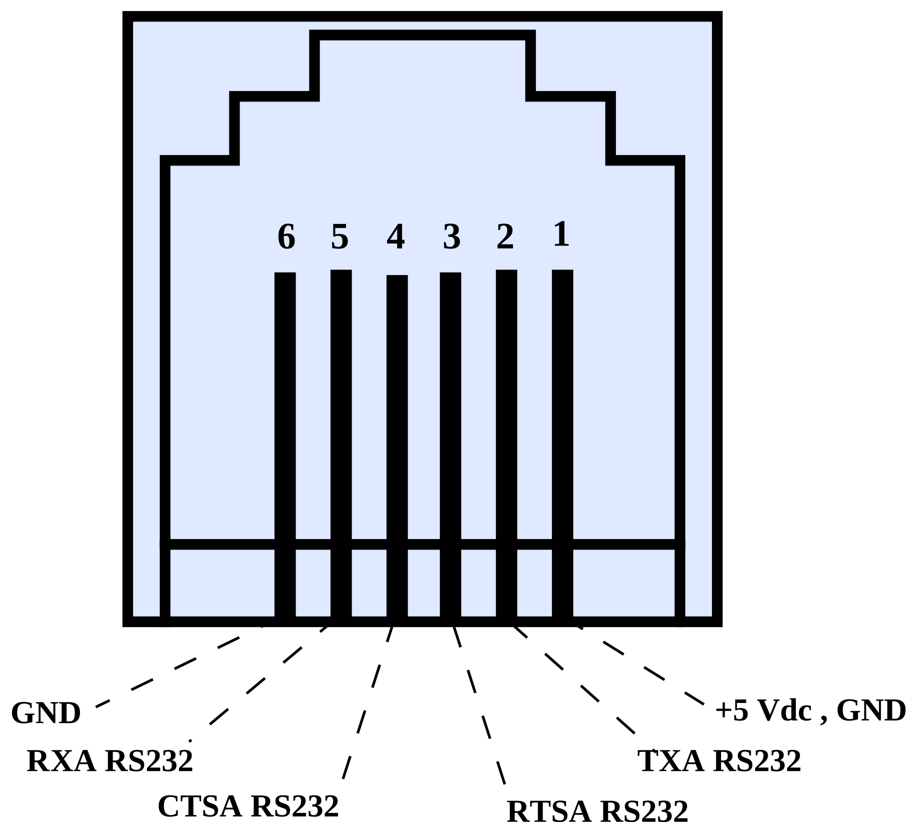


FIGURA 4: CN3A - CONNETTORE PER LINEA SERIALE A

Legenda:

RxA RS 232 = I - Receive Data: linea di ricezione in RS 232 della seriale A=ASP1.
TxA RS 232 = O - Transmit Data: linea di trasmissione in RS 232 della seriale A=ASP1.
CTSA RS 232 = I - Clear To Send: linea di abilitazione alla trasmissione in RS 232 della seriale A=ASP1.
RTSA RS 232 = O - Request To Send: linea di richiesta di trasmissione in RS 232 della seriale A=ASP1.
+5 Vdc,GND = I - Linea di alimentazione a +5 Vdc o linea di massa.
GND = - Linea di massa.

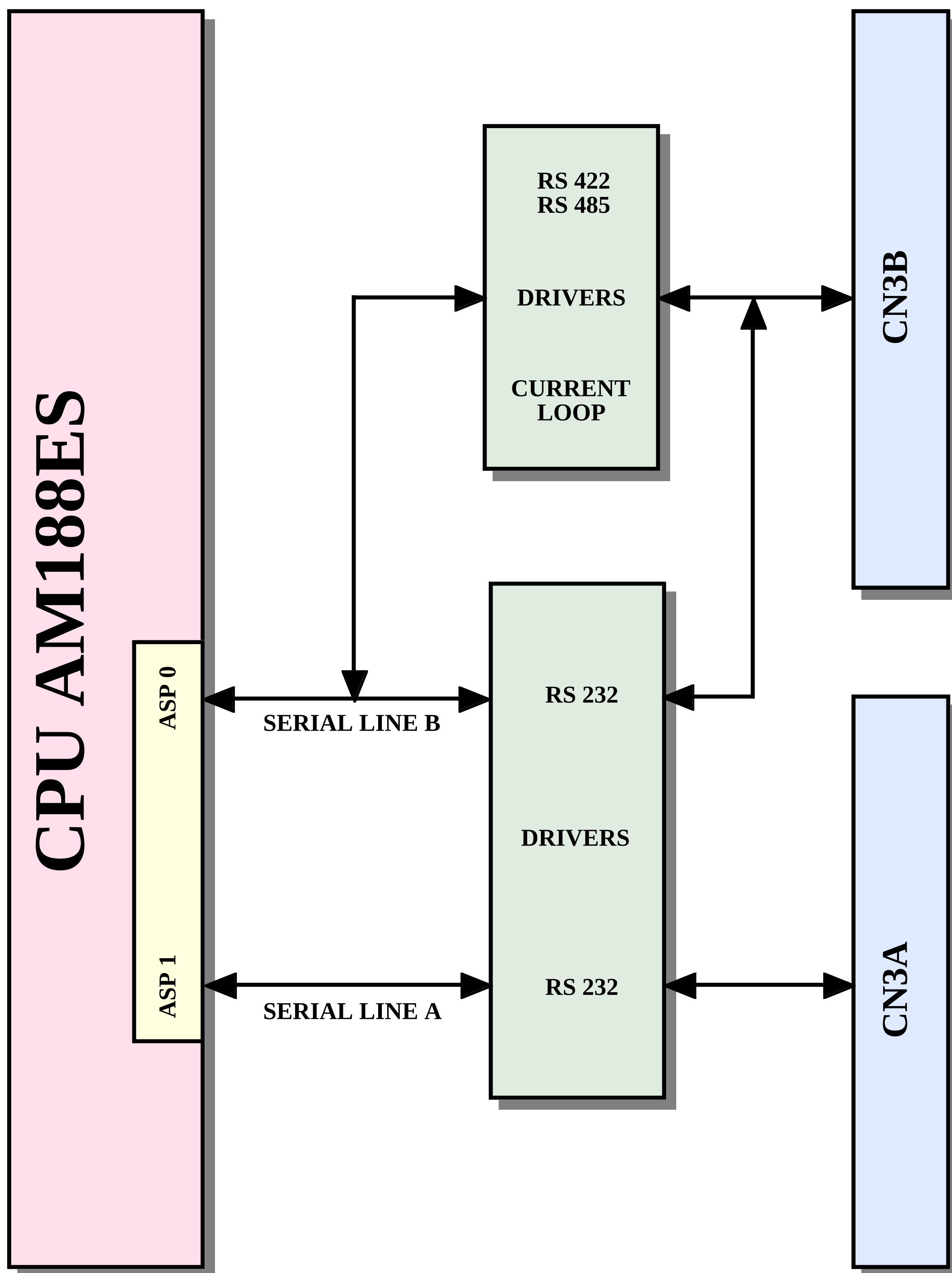


FIGURA 5: SCHEMA DI COMUNICAZIONE SERIALE

CN3B - CONNETTORE PER LINEA SERIALE B

CN3B é un connettore femmina, a 90 gradi,, del tipo plug a 6 vie.
Sul connettore sono disponibili i segnali per la comunicazione della linea seriale B, in RS 232, RS 422, RS 485 o current loop che é fisicamente collegata alla sezione ASP0 della CPU. La disposizione dei segnali, riportata di seguito, é stata studiata in modo da ridurre al minimo le interferenze ed in modo da facilitare la connessione con il campo, mentre i segnali rispettano le normative definite dal CCITT relative allo standard utilizzato.

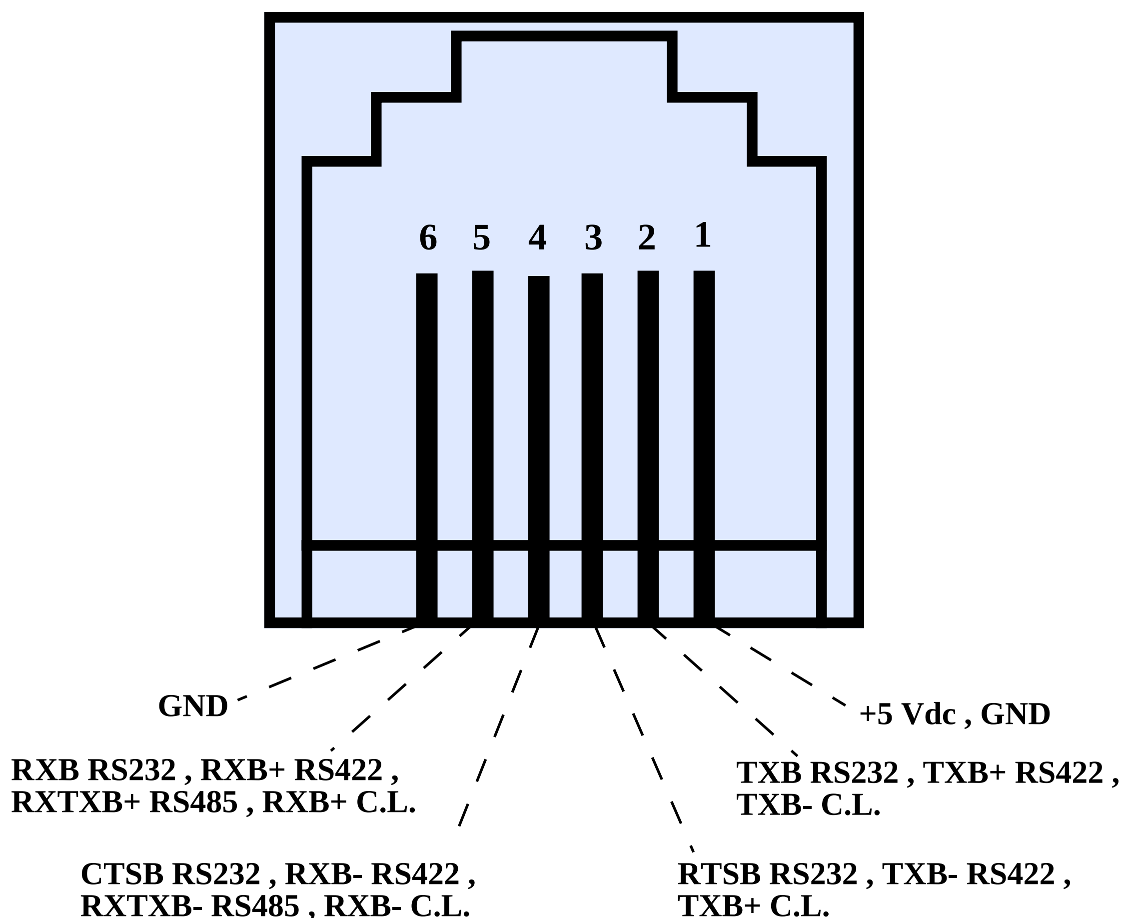


FIGURA 6: CN3B - CONNETTORE PER LINEA SERIALE B

Legenda:

RXB RS232	= I	- Receive Data: linea ricezione in RS 232 della seriale B=ASP0.
TXB RS232	= O	- Transmit Data: linea trasmissione in RS 232 della seriale B=ASP0.
CTSB RS232	= I	- Clear To Send: linea di abilitazione alla trasmissione in RS 232 della seriale B=ASP0.
RTSB RS232	= O	- Request To Send: linea di richiesta di trasmissione in RS 232 della seriale B=ASP0.
RXB- RS422	= I	- Receive Data Negative: linea bipolare negativa di ricezione differenziale in RS 422 della seriale B=ASP0.
RXB+ RS422	= I	- Receive Data Positive: linea bipolare positiva di ricezione differenziale in RS 422 della seriale B=ASP0.
TXB- RS422	= O	- Transmit Data Negative: linea bipolare negativa di trasmissione differenziale in RS 422 della seriale B=ASP0.

TXB+ RS422 = O - Transmit Data Positive: linea bipolare positiva di trasmissione differenziale in RS 422 della seriale B=ASP0.
RXTXB- RS485 = I/O - Receive Transmit Data Negative: linea bipolare negativa di ricezione e trasmissione differenziale in RS 485 della seriale B=ASP0.
RXTXB+ RS485 = I/O - Receive Transmit Data Positive: linea bipolare positiva di ricezione e trasmissione differenziale in RS 485 della seriale B=ASP0.
RXB- C.L. = I - Receive Data Negative: linea bipolare negativa di ricezione in Current Loop della seriale B=ASP0.
RXB+ C.L. = I - Receive Data Positive: linea bipolare positiva di ricezione in Current Loop della seriale B=ASP0.
TXB- C.L. = O - Transmit Data Negative: linea bipolare negativa di trasmissione in Current Loop della seriale B=ASP0.
TXB+ C.L. = O - Transmit Data Positive: linea bipolare positiva di trasmissione in Current Loop della seriale B=ASP0.
+5 Vdc = O - Linea di alimentazione a +5 Vdc.
GND = - Linea di massa.

CN2 - CONNETTORE PER BATTERIA ESTERNA DI BACK UP

CN2 é un connettore a scatolino, verticale, maschio, con passo 2,54 mm a 2 vie.
Tramite CN2 deve essere collegata una batteria esterna che provvede a mantenere i dati della RAM di bordo ed a garantire il funzionamento del real time clock, in assenza di tensione di alimentazione (per maggiori informazioni fare riferimento al paragrafo “BACK UP”.

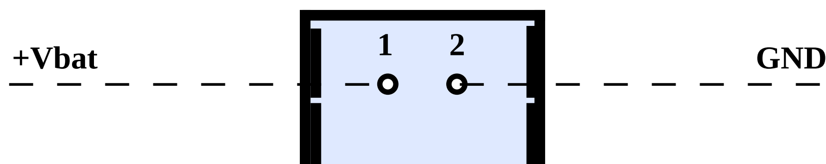


FIGURA 7: CN2 - CONNETTORE PER BATTERIA ESTERNA DI BACK UP

Legenda:

+Vbat = I - Positivo della batteria esterna di back up.
GND = - Negativo della batteria esterna di back up.

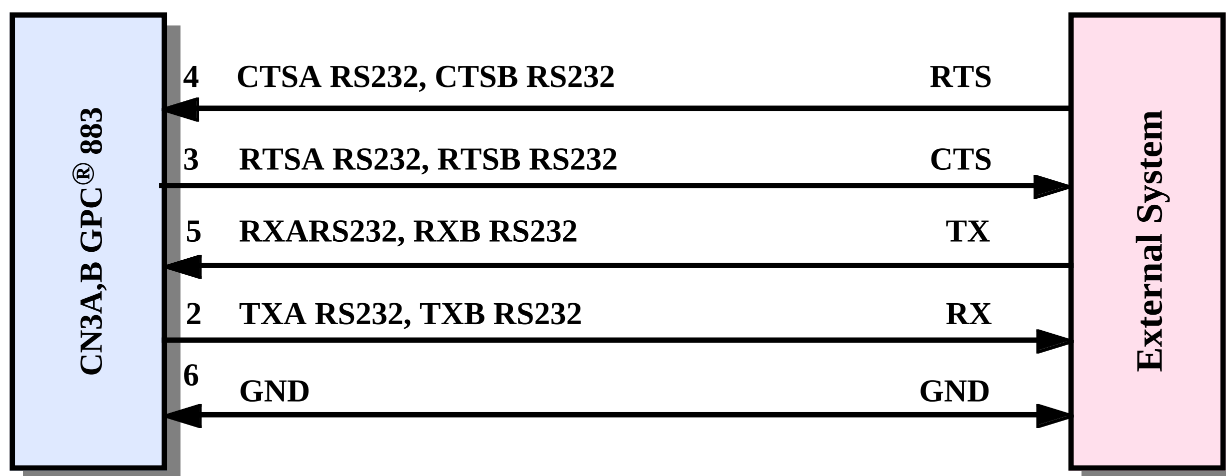


FIGURA 8: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN RS 232

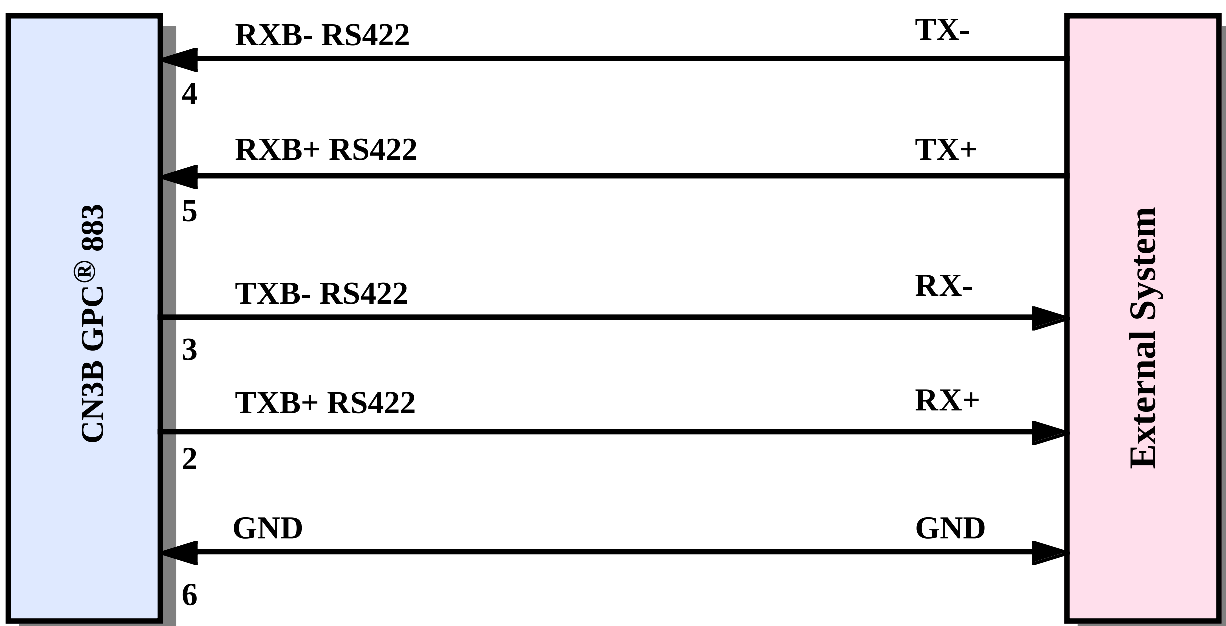


FIGURA 9: ESEMIPO COLLEGAMENTO PUNTO PUNTO IN RS 422

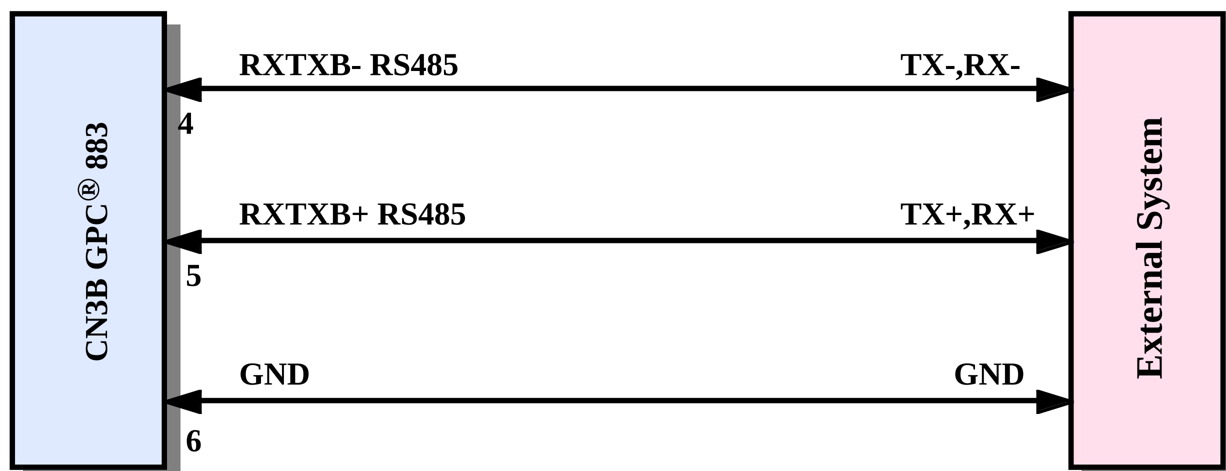


FIGURA 10: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN RS 485

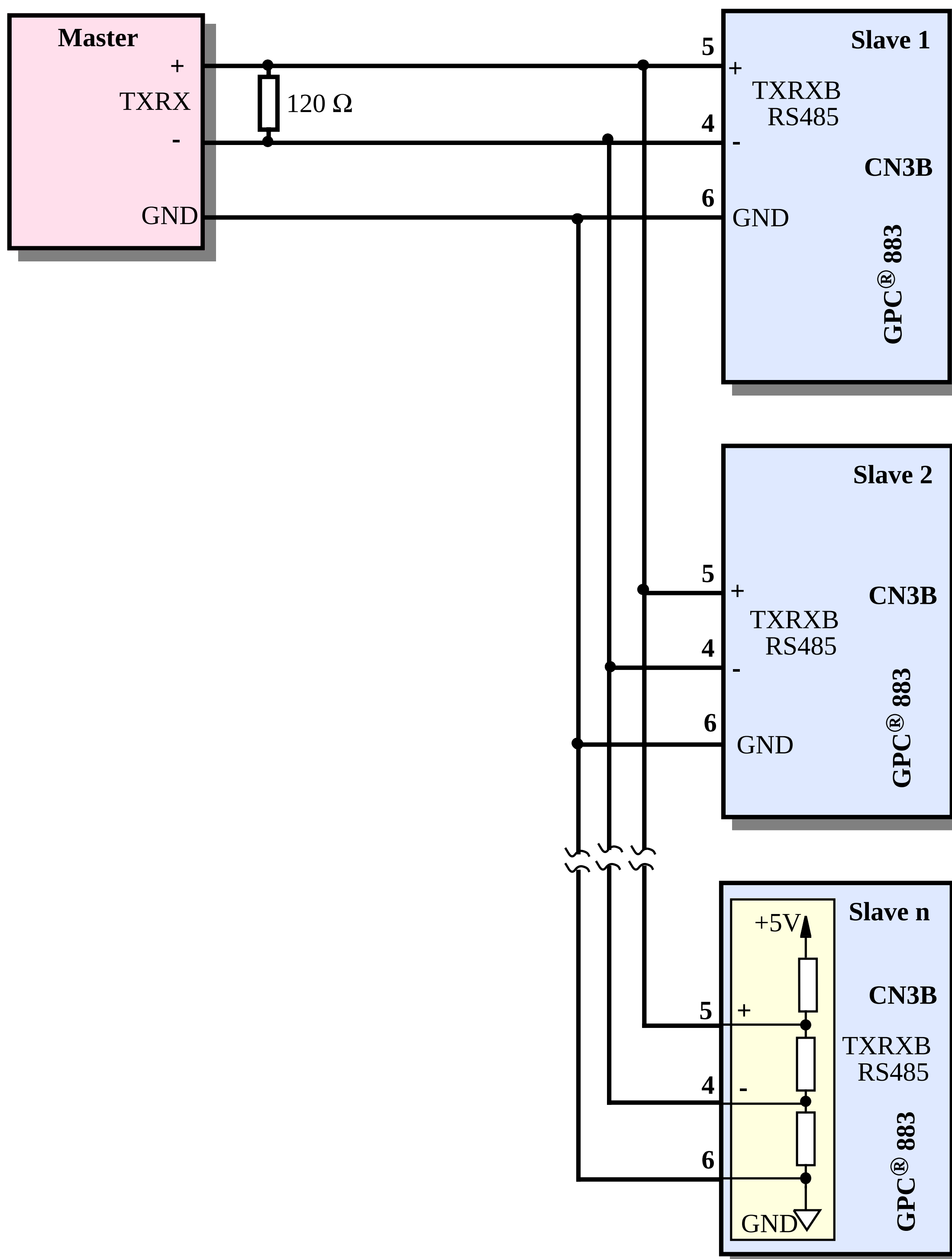


FIGURA 11: ESEMPIO COLLEGAMENTO IN RETE IN RS 485

Da notare che in una rete RS 485, devono essere presenti due resistenze di forzatura lungo la linea e due resitenze di terminazione (120 Ω), alle estremità della stessa, rispettivamente vicino all'unità Master ed all'ultima unità Slave.
A bordo della **GPC® 883** è presente la circuiteria di terminazione e forzatura, che può essere inserita o disinserita, tramite appositi jumpers, come illutrato in seguito.
In merito alla resistenza di terminazione dell'unità Master, provvedere a collegarla solo se questa non é già presente al suo interno (ad esempio molti convertitori RS232-RS485 ne sono già provvisti).
Per maggiori informazioni consultare il Data-Book TEXAS INSTRUMENTS, "*RS 422 and RS 485 Interface Cicuits*", nella parte introduttiva riguardante le reti RS 422-485.

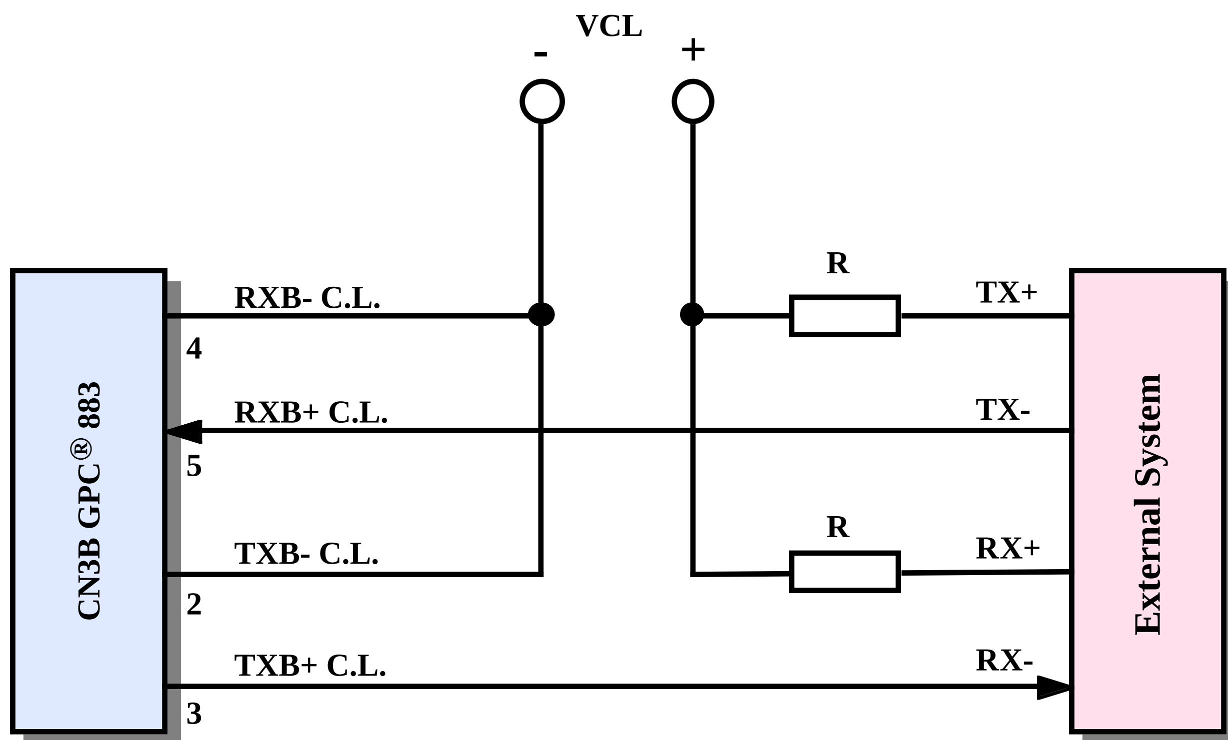


FIGURA 12: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN CURRENT LOOP A 4 FILI

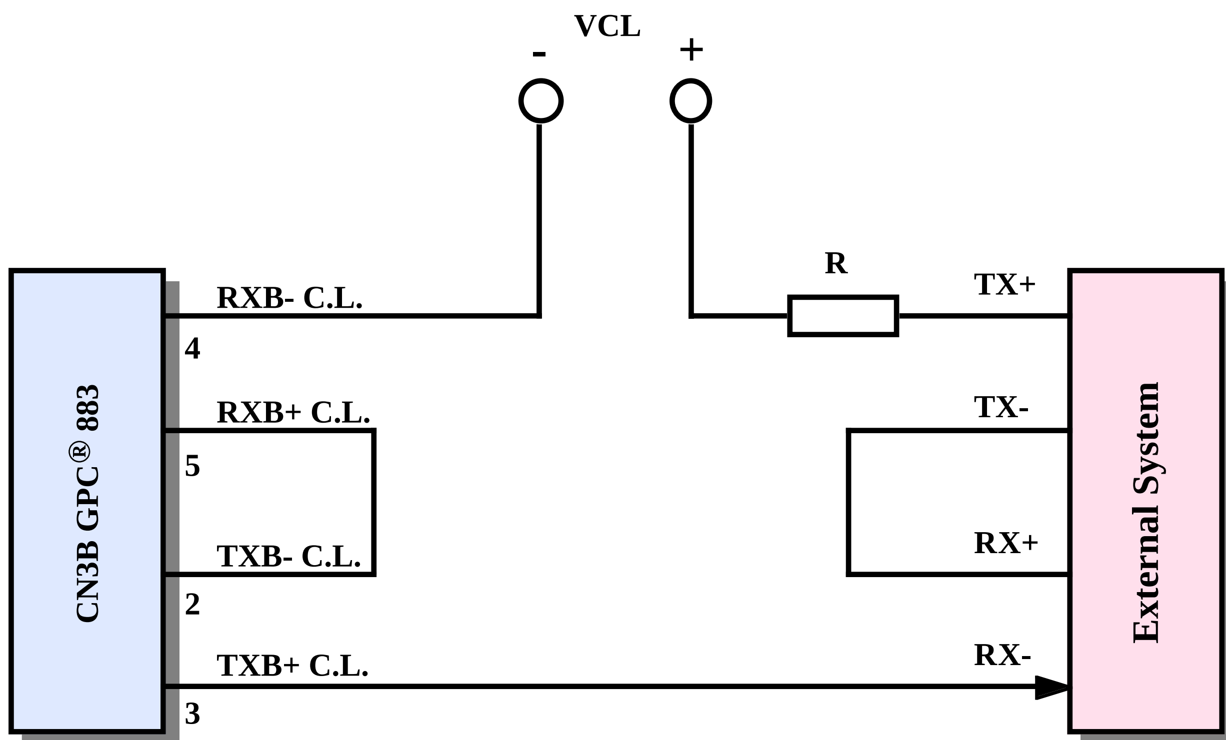


FIGURA 13: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN CURRENT LOOP A 2 FILI

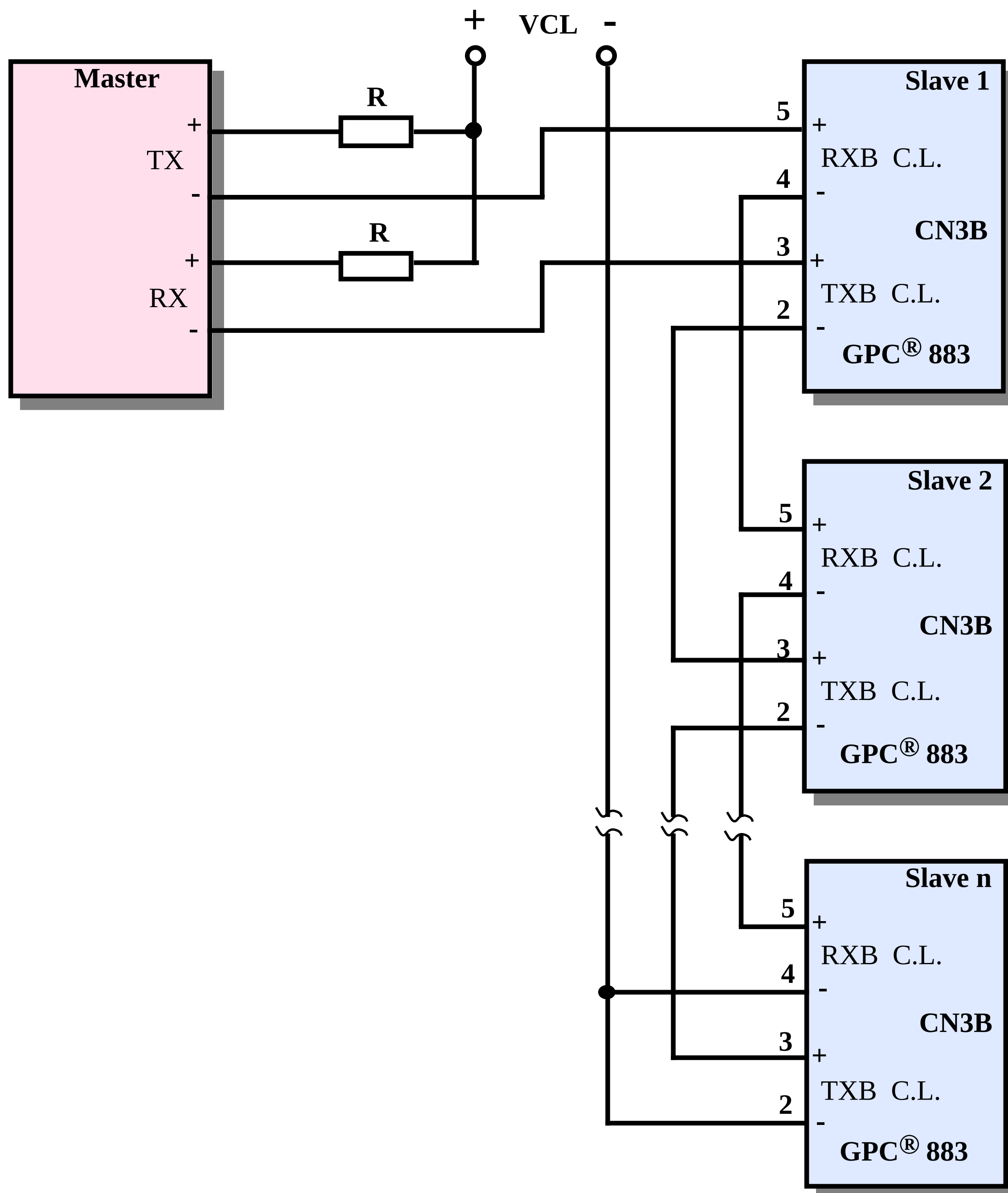


FIGURA 14: ESEMPIO DI COLLEGAMENTO IN RETE IN CURRENT LOOP

Per il collegamento in current loop passivo sono possibili due diversi tipi di collegamento: a 2 fili ed a 4 fili. Tali connessioni sono riportate nelle figure 12÷14; in esse é indicata la tensione per alimentare l'anello (**VCL**) e le resistenze di limitazione della corrente (**R**). I valori di tali componenti variano in funzione del numero di dispositivi collegati e della caduta sul cavo di collegamento; bisogna quindi effettuare la scelta considerando che:

- si deve garantire la circolazione di una corrente di **20 mA**;
- su ogni trasmettitore cadono mediamente **2,35 V** con una corrente di 20 mA;
- su ogni ricevitore cadono mediamente **2,52 V** con una corrente di 20 mA;
- in caso di cortocircuito sulla rete ogni trasmettitore dissipi al massimo **125 mW**;
- in caso di cortocircuito sulla rete ogni ricevitore dissipi al massimo **90 mW**.

Per maggiori informazioni consultare il Data-Book HEWLETT-PACKARD, nella parte che riguarda gli opto accoppiatori per current loop denominati **HCPL 4100** e **HCPL 4200**.

CN5 - CONNETTORE PER I/O DEL PIO E PPI

CN5 è un connettore a scatolino verticale, maschio, con passo 2.54 mm, a 20 piedini.
Tramite CN5 si effettua la connessione tra le 18 linee di I/O digitale delle interfaccie periferiche programmabili PIO e PPI e l'ambiente esterno. Alcuni piedini di questo connettore hanno una duplice funzione infatti, via software, alcune sezioni interne della CPU possono essere multiplexate con i segnali di I/O. Tutti i segnali di CN5 coincidono con segnali logici a livello TTL e seguono il pin out standardizzato I/O **ABACO®**.

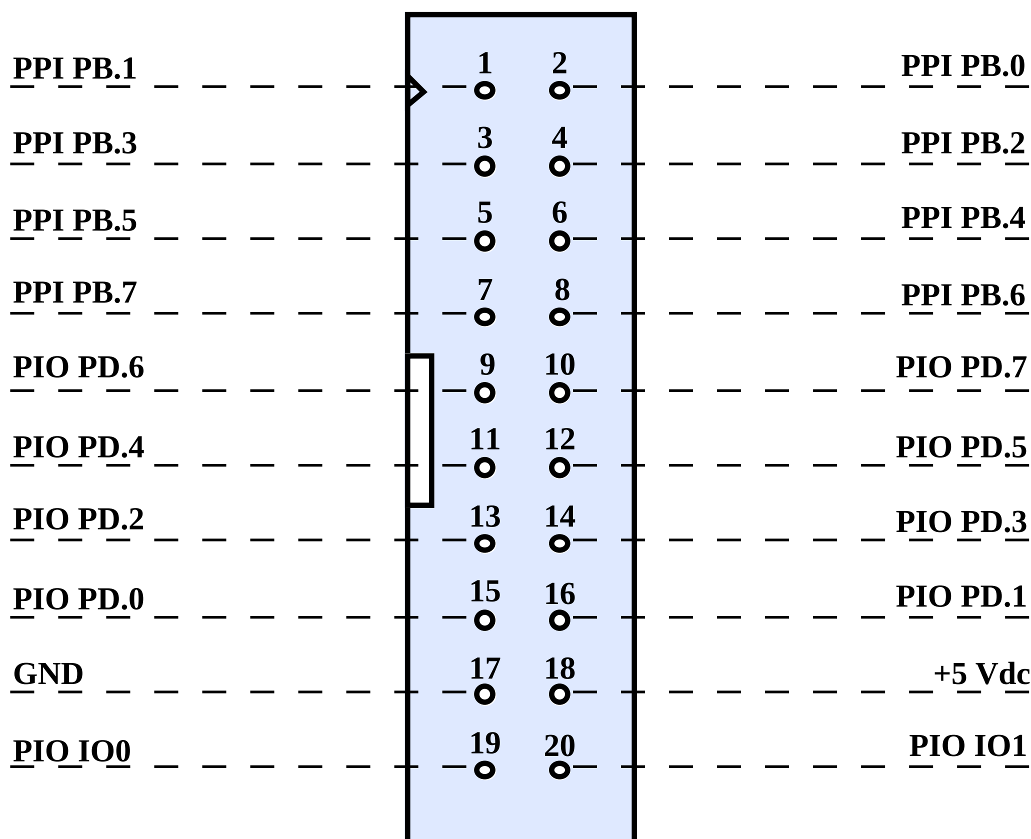


FIGURA 15: CN5 - CONNETTORE PER I/O DEL PIO E PPI

Legenda:

PPI PB.n	= I/O - Linea digitale n del port B del PPI 82C55.
PIO PD.n	= I/O - Linee digitali del PIO (vedere paragrafo "PIN MULTIFUNZIONE").
PIO IO0	= I/O - Linea digitale del PIO (vedere paragrafo "PIN MULTIFUNZIONE").
PIO IO1	= I/O - Linea digitale del PIO (vedere paragrafo "PIN MULTIFUNZIONE").
+5 Vdc	= O - Linea di alimentazione a +5 Vdc.
GND	= - Linea di massa.

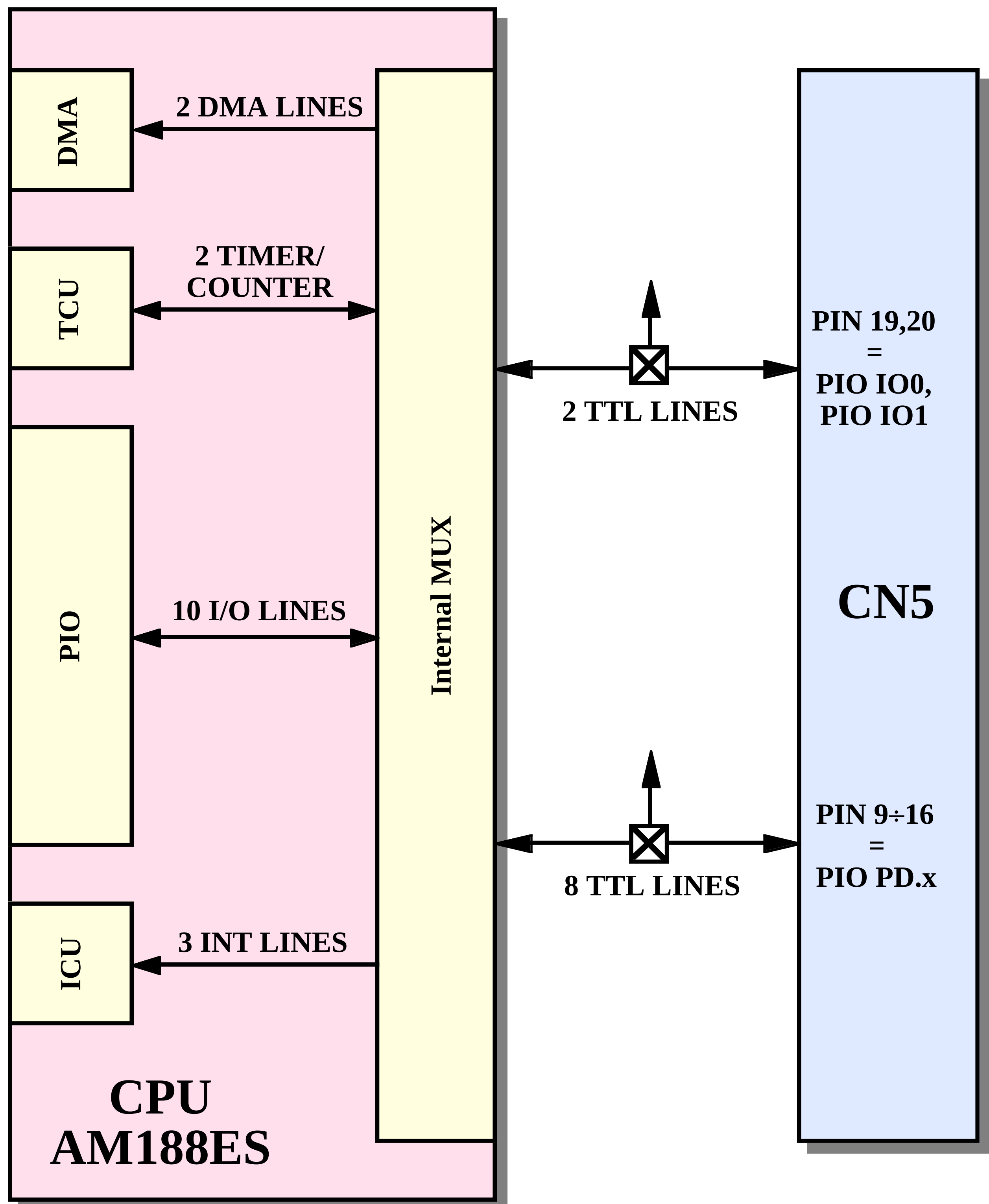


FIGURA 16: SCHEMA DI COLLEGAMENTO I/O DEL PIO

CN7 - CONNETTORE PER I/O DEL PPI

CN7 è un connettore a scatolino verticale con passo 2.54 mm a 20 piedini.
Tramite CN7 si effettua la connessione tra l'interfaccia periferica programmabile PPI 82C55 e l'ambiente esterno, utilizzando due dei tre port paralleli ad 8 bit di cui dispone. I segnali presenti su questo connettore coincidono con segnali logici a livello TTL e seguono il pin out standardizzato I/O **ABACO®**.

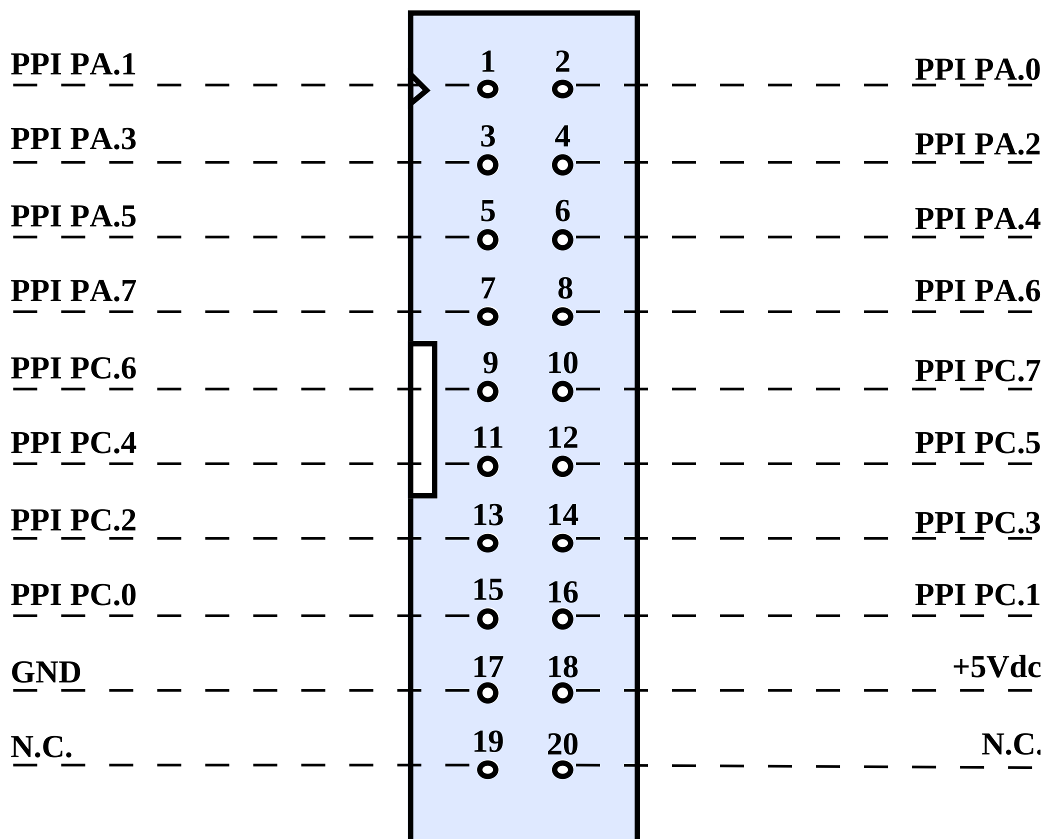


FIGURA 17: CN7 - CONNETTORE PER I/O DEL PPI

Legenda:

PA.n = I/O - Linea digitale n del port A del PPI 82C55.
PC.n = I/O - Linea digitale n del port C del PPI 82C55.
+5 Vdc = O - Linea di alimentazione a +5 Vdc
GND = - Linea di massa
N.C. = - Non Collegato

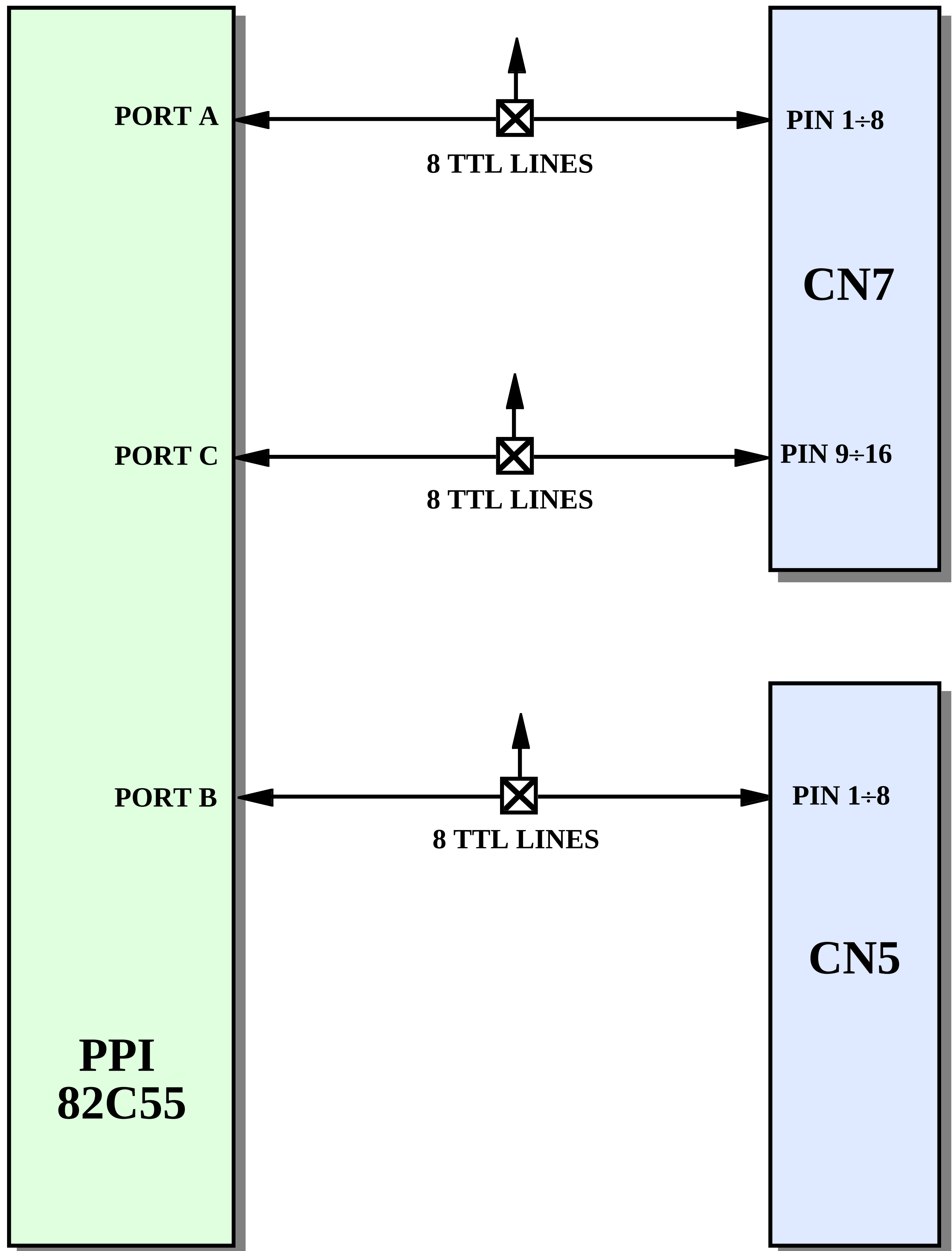


FIGURA 18: SCHEMA DEL COLLEGAMENTO I/O DEL PPI

CN6 - CONNETTORE PER INGRESSI A/D CONVERTER

CN6 é un connettore a scatolino, verticale, con passo 2.54 mm a 20 piedini.
Tramite CN6 possono essere collegate le 8 linee analogiche d'ingresso con il campo esterno. Tali linee sono direttamente collegate all'A/D di bordo della scheda, sono a bassa impedenza, sono provviste di un condensatore di filtro e possono variare nei range 0÷5 V, 0÷10 V, ±5 V, ±10 V. Tramite l'installazione di un opportuno modulo di conversione (**.8420**) é inoltre possibile collegare agli 8 ingressi dei segnali in corrente nel range 0÷20 mA o 4÷20 mA. La disposizione dei segnali su questo connettore é studiata in modo da ridurre tutti i problemi di rumore ed interferenza, garantendo quindi un'ottima trasmissione del segnale.

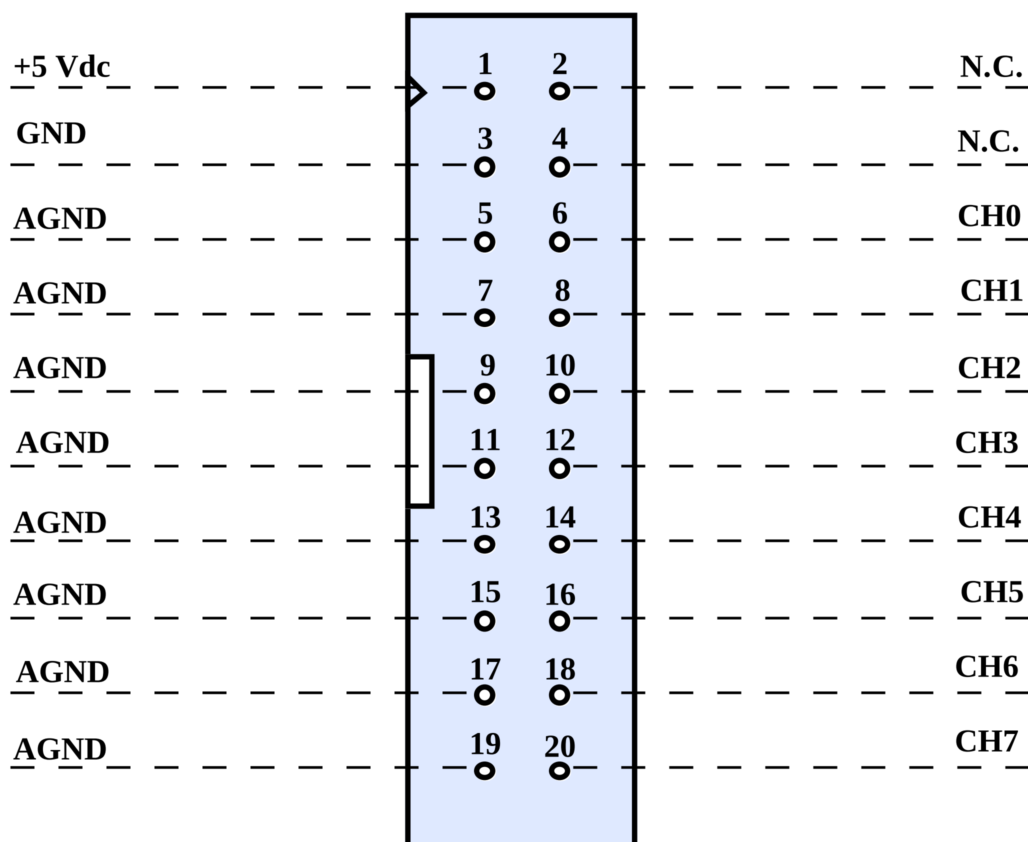


FIGURA 19: CN6 - CONNETTORE PER INGRESSI A/D CONVERTER

Legenda:

CHn	=	I	- Linea analogica d'ingresso collegata al canale n dell'A/D converter
AGND	=		- Linea di massa analogica
+5 Vdc	=	O	- Linea di alimentazione a +5 Vdc
GND	=		- Linea di massa
N.C.	=		- Non collegato

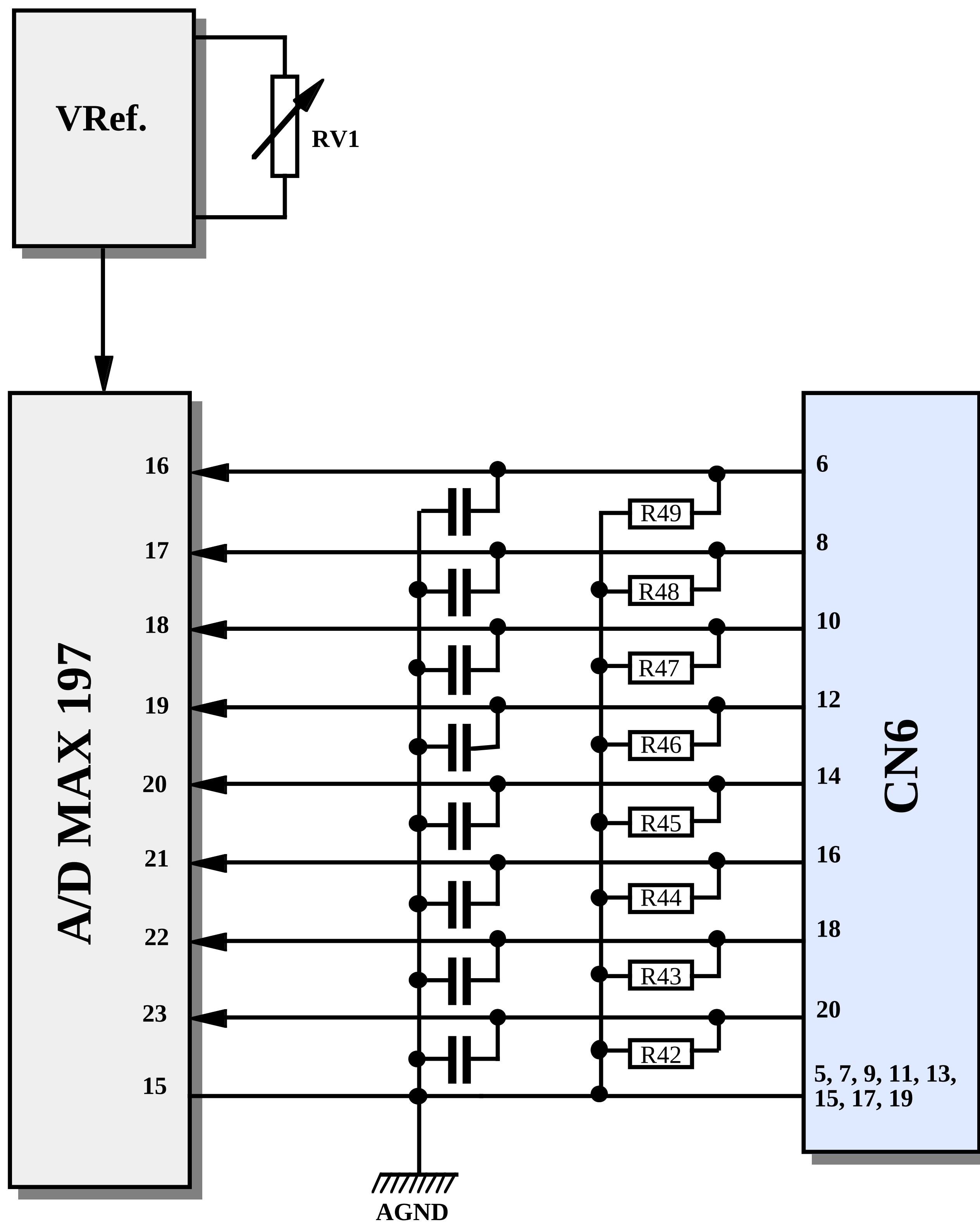


FIGURA 20: SCHEMA DI COLLEGAMENTO A/D CONVERTER

CN10 - CONNETTORE PER LINEA CAN

CN10 é un connettore a vaschetta D a 9 vie maschio, a 90°.
Tramite CN10 si può collegare la scheda ad una linea di comunicazione seriale CAN ottenendo un veloce, comodo ed efficiente nodo sul BUS di campo definito dallo stesso protocollo. La disposizione dei segnali é stata studiata in modo da ridurre al minimo le interferenze ed in modo da facilitarela connessione con il campo, seguendo le normative dello standard CIA DS102.

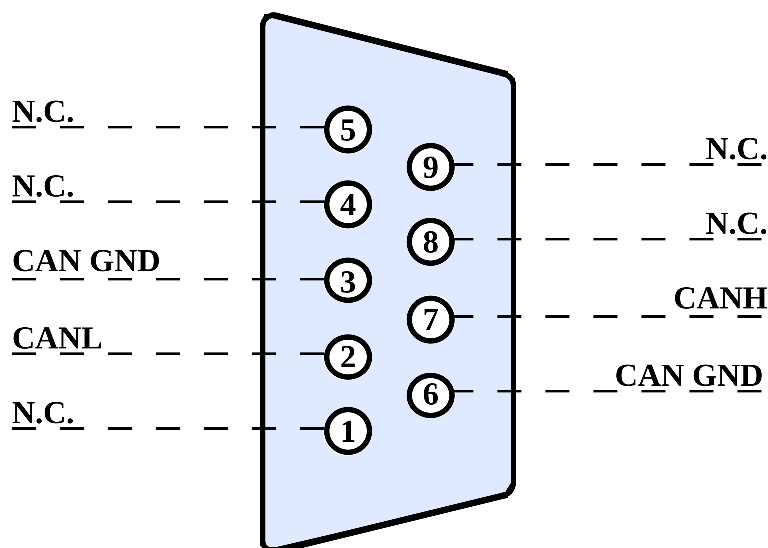


FIGURA 21: CN10 - CONNETTORE PER LINEA CAN

Legenda:

CANH	= I/O	- Linea differenziale high per CAN BUS.
CANL	= I/O	- Linea differenziale low per CAN BUS.
CAN GND	=	- Linea di massa della linea CAN.
N.C.	=	- Non connesso.

N.B.
La tensione di alimlentazione della sezione CAN é galvanicamente isolata dalla tensione di alimentazione della **GPC® 883** quindi il segnale CAN GND non deve essere collegato ai segnali GND ed AGND presenti sugli altri connettori. Per ulteriori informazioni consultare il paragrafo "ALIMENTAZIONE".

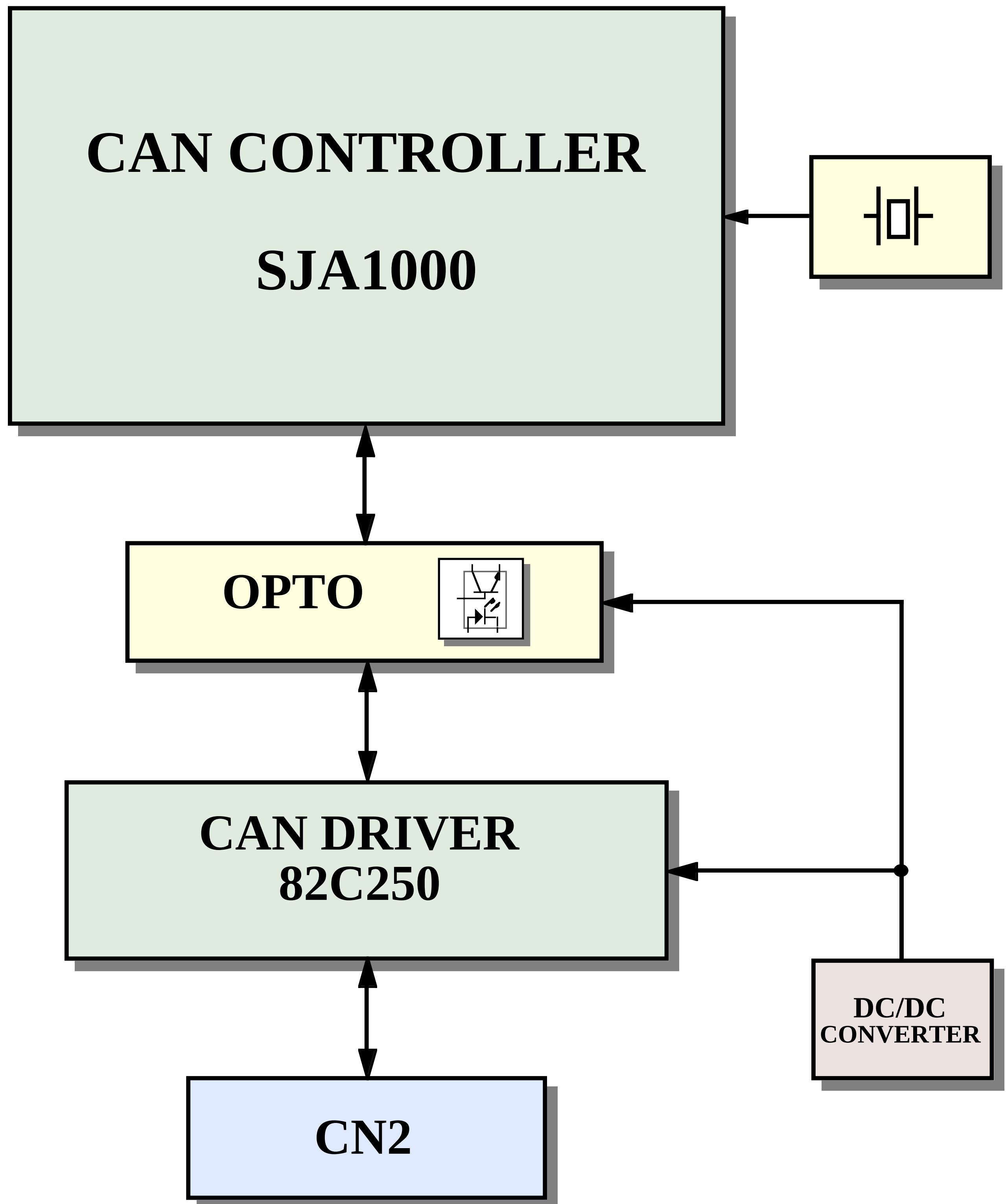


FIGURA 22: SCHEMA DI COLLEGAMENTO LINEA CAN

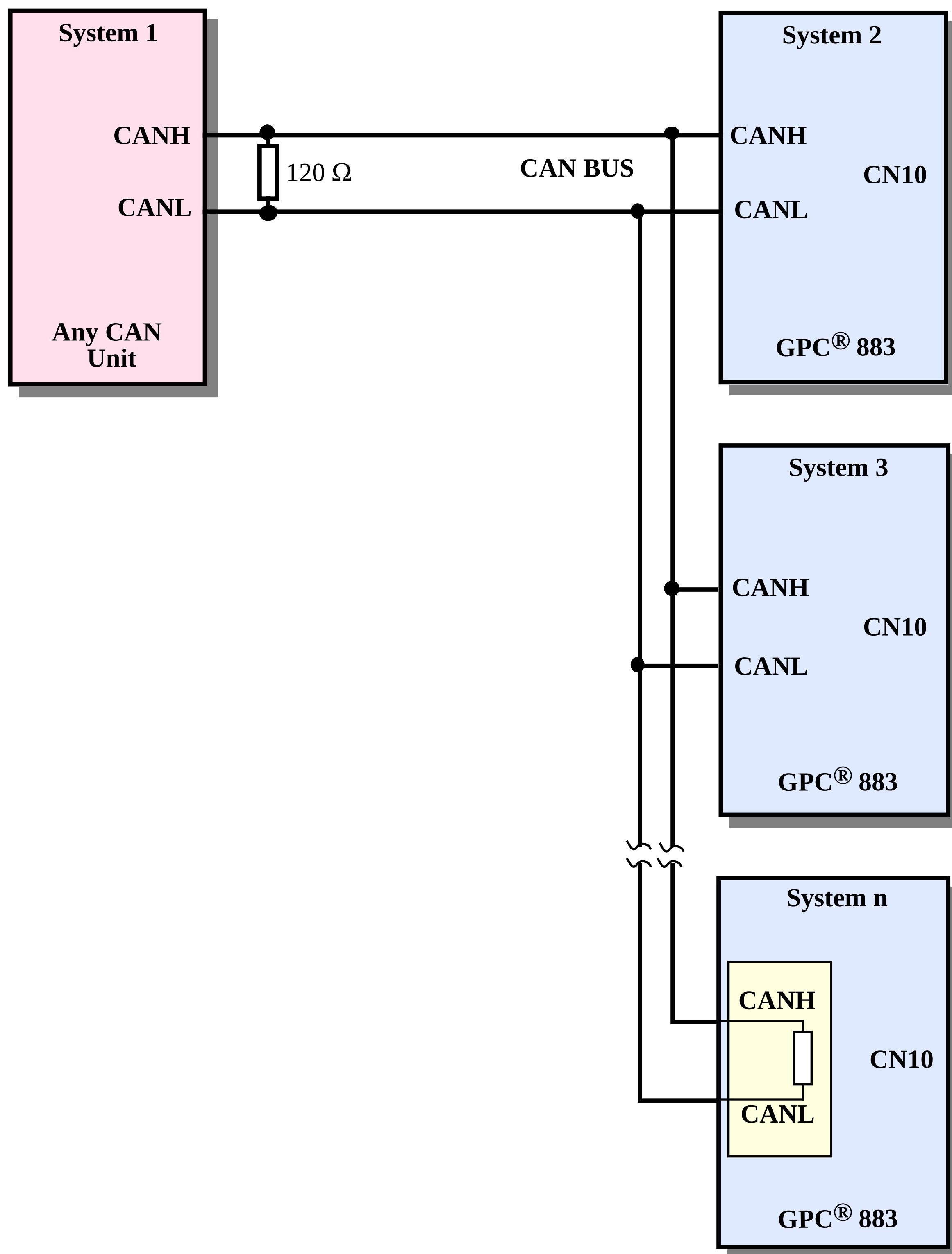


FIGURA 23: ESEMPIO COLLEGAMENTO IN RETE CON BUS CAN

Da notare che una rete CAN, deve avere un'impedenza di linea di 60 Ω e per questa ragione lungo la linea possono essere presenti due resitenze di terminazione (120 Ω), alle estremità della stessa. A bordo della **GPC® 883** è presente la circuiteria di terminazione che può essere inserita o disinserita, tramite un apposito jumper, come illutrato in seguito.
Qualora i sistemi collegati sulla rete CAN risultino a differenze di potenziale elevate si può ovviare ad eventuali problemi di comunicazione e/o funzionamento, collegando anche le masse dei sistemi ovvero i pin 3 e 6 di CN10.

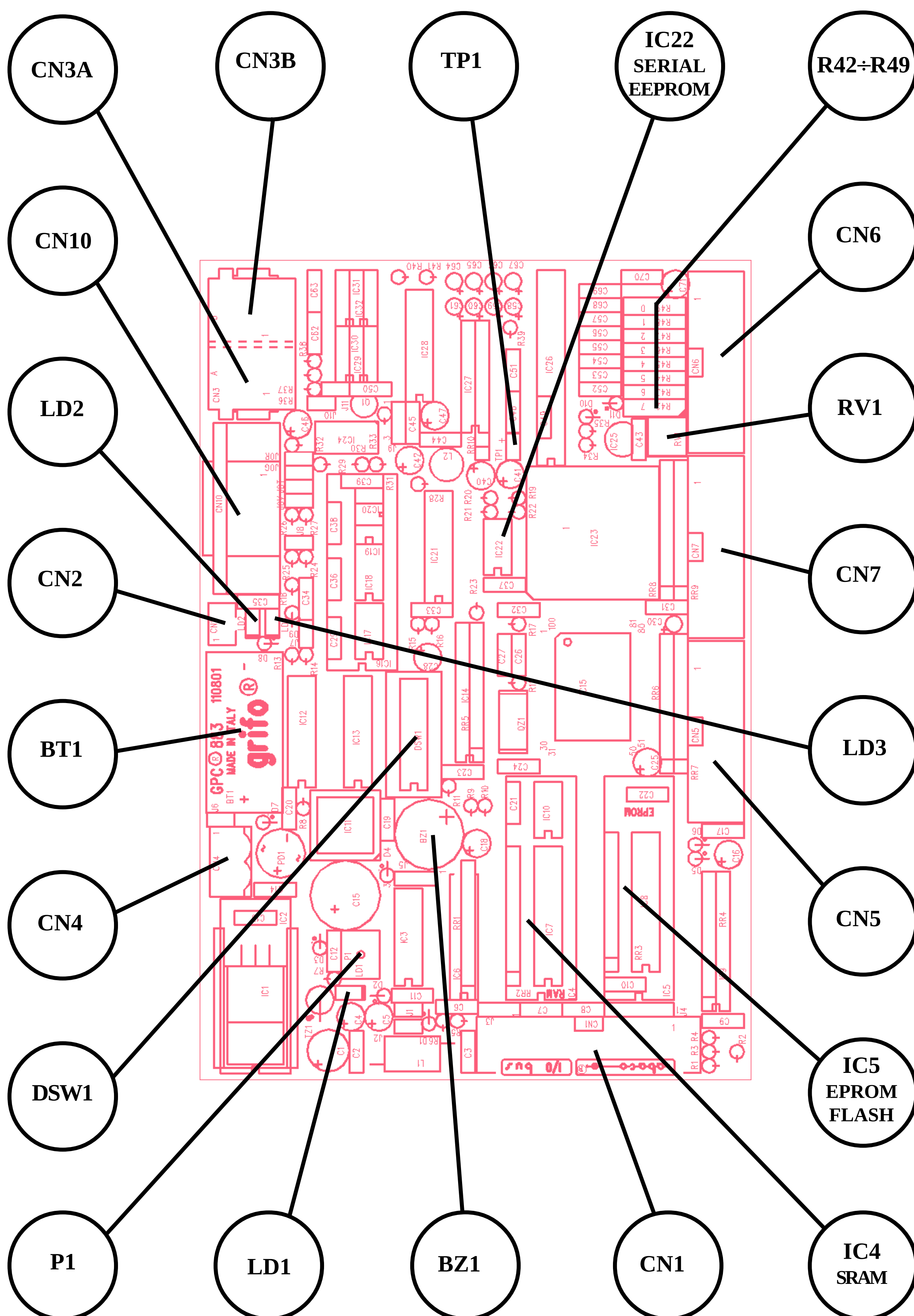


FIGURA 24: DISPOSIZIONE CONNETTORI, TRIMMER, BATTERIA, MEMORIE, ECC.

INTERFACCIAMENTO CONNETTORI CON IL CAMPO

Al fine di evitare eventuali problemi di collegamento della scheda con tutta l'elettronica del campo a cui la **GPC® 883** si deve interfacciare, si devono seguire le informazioni riportate nei precedenti paragrafi e le relative figure che illustrano le modalità interne di connessione.

- Per i segnali che riguardano la comunicazione seriale con i protocolli RS 232, RS 422, RS 485, current loop e CAN fare riferimento alle specifiche standard di ognuno di questi protocolli.
- Tutti i segnali a livello TTL possono essere collegati a linee dello stesso tipo riferite alla massa digitale della scheda. Il livello 0V corrisponde allo stato logico 0, mentre il livello 5V corrisponde allo stato logico 1.
- I segnali d'ingresso alla sezione A/D devono essere collegati a segnali analogici a bassa impedenza che rispettino il range di variazione ammesso che può arrivare a ±10 V a seconda della programmazione software oppure 0÷20 mA. Da notare che gli 8 ingressi analogici presenti su CN6 sono dotati di condensatori di filtro che garantiscono una maggiore stabilità sul segnale acquisito, ma che allo stesso tempo abbassano la frequenza di taglio (1 MHz che é comunque superiore alla massima frequenza di acquisizione A/D=110 KHz).
- Per evitare problemi di pilotaggio, al connettore CN1 per **ABACO®** I/O BUS possono essere collegate fino ad un massimo di tre schede periferiche. Inoltre per minimizzare l'effetto di eventuali disturbi dal campo, la lunghezza del flat cable a 26 vie di interconnessione non dovrebbe superare i 250 mm di lunghezza.

TRIMMER E TARATURE

Sulla **GPC® 883** é presente il trimmer RV1 utilizzato per la taratura della scheda; tale componente permette di fissare il valore della tensione di riferimento su cui si basa la sezione di A/D Converter. La scheda viene sottoposta ad un accurato test di collaudo che provvede a verificare la funzionalità della stessa ed allo stesso tempo a tararla in tutte le sue parti. La taratura viene effettuata in laboratorio a temperatura costante di +20 gradi centigradi, seguendo la procedura di seguito descritta:

- Si effettua la taratura di precisione della Vref della sezione A/D ad un valore di 4,096 V regolando il trimmer RV1, tramite un multimetro galvanicamente isolato a 5 cifre collegato al test point TP1.
- Si verifica la corrispondenza tra segnale analogico fornito in ingresso e combinazione letta dalla sezione A/D converter. La verifica viene effettuata fornendo un segnale di verifica con un calibratore campione e controllando che la differenza tra la combinazione determinata dalla scheda e quella determinata in modo teorico, non superi la somma degli errori della sezione A/D.
- Si blocca il trimmer della scheda, opportunamente tarato, tramite vernice.

Le sezioni d'interfaccia analogica utilizzano componenti di alta precisione che vengono addirittura scelti in fase di montaggio, proprio per evitare lunghe e complicate procedure di taratura. Per questo una volta completato il test di collaudo e quindi la taratura, il trimmer RV1 viene bloccato, in modo da garantire una immunità della taratura anche ad eventuali sollecitazioni meccaniche (vibrazioni, spostamenti, ecc.).
L'utente di norma non deve intervenire sulla taratura della scheda, ma se lo dovesse fare (a causa di derive termiche, derive del tempo, ecc.) deve rigorosamente seguire la procedura sopra illustrata.
Per una facile individuazione di RV1 e TP1 a bordo scheda, si faccia riferimento alla figura 24.

SELEZIONE TIPO INGRESSI ANALOGICI

La scheda **GPC® 883**, può avere ingressi analogici in tensione e/o corrente, come descritto nei precedenti paragafi e capitoli. La selezione del tipo d'ingresso viene essere effettuata in fase di ordine della scheda montando un apposito modulo opzionale di conversione corrente-tensione basato su resistenze di caduta di precisione (codice opzione **.8420**). In particolare vale la corrispondenza:

R49	->	canale 0
R48	->	canale 1
R47	->	canale 2
R46	->	canale 3
R45	->	canale 4
R44	->	canale 5
R43	->	canale 6
R42	->	canale 7

Nel caso il modulo corrente-tensione non sia montato (default) il corrispondente canale accetta un ingresso in tensione nei range 0÷5 V, 0÷10 V, ±5 V o ±10V (selezionabili via software, in modo indipendente per ogni canale); viceversa un ingresso in corrente.
In questo caso il canale in questione deve essere configurato in modalità **0÷5 V**, ne segue che il valore della resistenza, su cui si basa il convertitore corrente-tensione, si ottiene dalla seguente formula:

R = 5 V / Imax

Normalmente i moduti di conversione tensione-corrente, si basano su resistenze di precisione da **248Ω**, relative ad ingressi 4÷20 mA o 0÷20 mA. Per una facile individuazione del modulo descritto e delle relative resistenze componenti, fare riferimento alla figura 24, mentre per esigenze al di fuori dei valori sopracitati si prega di contattare la **grifo®**.

TEST POINT

La scheda é provvista di un test point denominato TP1, che permette la lettura attraverso un multimetro galvanicamente isolato, della tensione di riferimento che viene regolata in laboratorio a Vref=4,096 V. Il TP1 é composto da due contatti con la seguente corrispondenza:

pin +	->	Vref
pin -	->	GND

Per una facile individuazione del test point si faccia riferimento alla figura 24, per il riconoscimento dei due pin si utilizzi la serigrafia della scheda, mentre per ulteriori informazioni sul segnale Vref si veda il paragrafo "TRIMMER E TARATURE".

INTERFACCIE PER I/O DIGITALI

Tramite CN5 e CN7 (connettori compatibili con standard di I/O **ABACO®**) si può collegare la **GPC® 883** ai numerosi moduli del carteggio **grifo®** che riportano lo stesso pin out. Dal punto di vista dell'installazione, queste interfaccie richiedono solo un flat cable da 20 vie intestato con due connettori da 20 vie (FLT.20+20) con cui é possibile portare anche le alimentazioni, mentre dal punto di vista software la gestione é altrettanto semplice ed immediata, infatti i pacchetti software disponibili per la **GPC® 883** sono provvisti di tutte le procedure necessarie. Quest'ultime per la maggioranza dei pacchetti software disponibili, coincidono con dei "driver software" o delle librerie aggiunti al linguaggio di programmazione, che consentono di utilizzare direttamente le istruzioni ad alto livello dello stesso linguaggio di programmazione e quindi tutta la loro potenza.
Di particolare interesse é la possibilità di collegare direttamente una serie di moduli come:

- **QTP 16P**, **QTP 24P**, **KDx x24**, **DEB 01**, ecc. con cui risolvere tutti i problemi di interfacciamento operatore locale. Questi moduli sono già dotati delle risorse necessarie per gestire un buon livello di colloquio uomo-macchina (includono infatti display alfanumerici, tastiera a matrice e LEDs di visualizzazione) ad una breve distanza dalla **GPC® 883**. Dal punto di vista software i driver disponibili rendono utilizzabili le risorse dell'interfaccia operatore direttamente con le istruzioni ad alto livello per la gestione della console.

- **MCI 64** con cui risolvere tutti i problemi di salvataggio di grosse quantità di dati. Questo modulo é dotato di un connettore per memory card PCMCIA su cui possono essere inserite vari tipi di memory card (RAM, FLASH, ROM, ecc) nei vari size disponibili. Dal punto di vista software i driver disponibili coincidono con un completo file system e rendono utilizzabili le memory card direttamente con le istruzioni ad alto livello per la gestione dei files.

- **IAC 01**, **DEB 01** con cui gestire una stampante con interfaccia parallela CENTRONICS. Quest'ultima può essere collegata direttamente all'interfaccia, con un cavo standard, e quindi gestita con le istruzioni relative alla stampante del linguaggio di programmazione utilizzato.

- **RBO xx**, **TBO xx**, **XBI xx**, **OBI xx** con cui bufferare i segnali di I/O TTL nei confronti del campo. Con questi moduli i segnali di input vengono convertiti in ingressi optoisolati di tipo NPN o PNP, mentre i segnali di output vengono convertiti in uscite galvanicamente isolate a transistor o relé.

Per maggiori informazioni relative alle interfaccie per I/O digitali si veda il capitolo "SCHEDE ESTERNE" e la documentazione del software utilizzato.

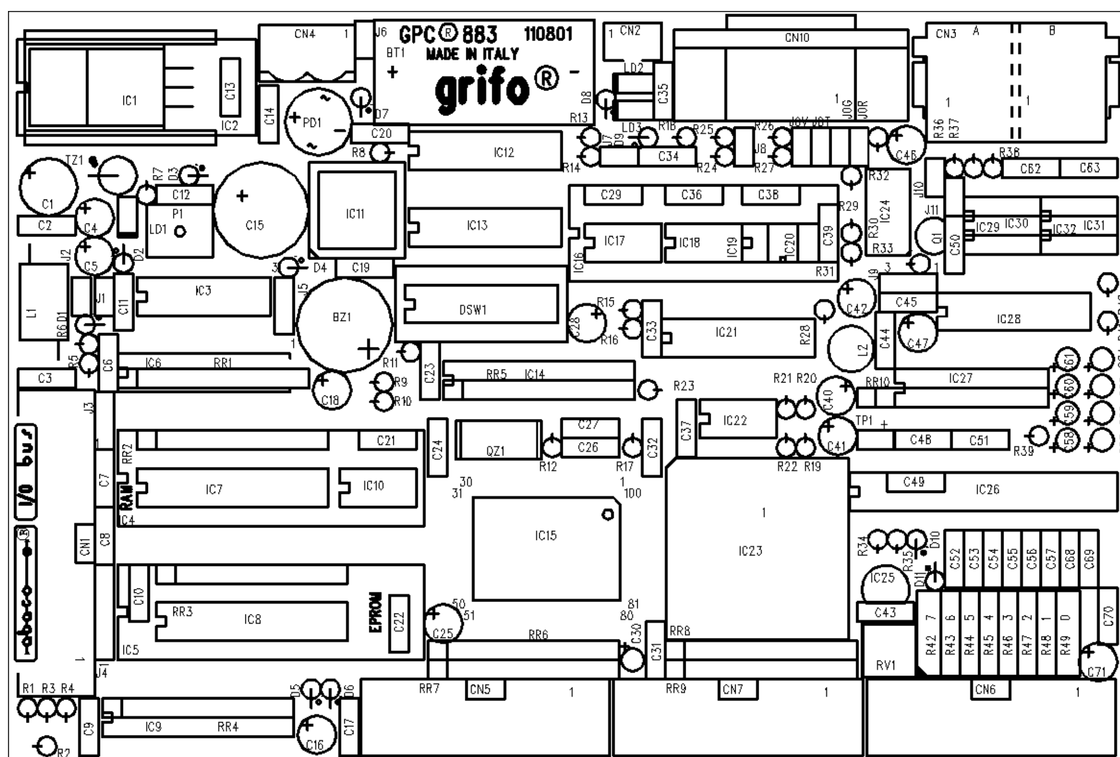


FIGURA 25: PIANTA COMPONENTI LATO COMPONENTI

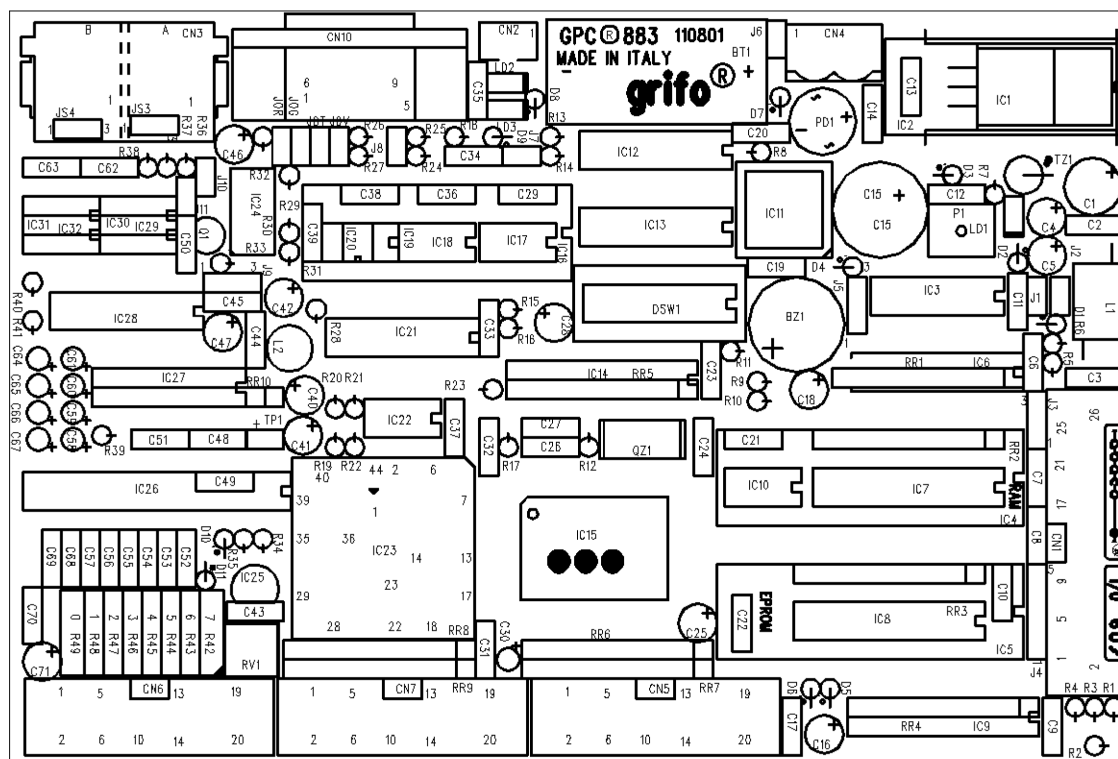


FIGURA 26: PIANTA COMPONENTI LATO STAGNATURE

JUMPERS

Esistono a bordo della **GPC® 883** 14 jumpers, di cui 11 a cavalliere e 3 a stagno, con cui é possibile effettuare alcune selezioni che riguardano il modo di funzionamento della stessa. Di seguito ne é riportato l'elenco, l'ubicazione e la loro funzione nelle varie modalità di connessione.

JUMPER	N. VIE	UTILIZZO
J1	2	Jumper di configurazione per selezionare modalità RUN o DEBUG
J2	2	Seleziona collegamento pin 26 di CN1=**ABACO®** I/O BUS
J3	3	Seleziona dimensioni dispositivo SRAM su IC4
J4	5	Seleziona tipo di memoria su IC5
J5	3	Seleziona collegamento circuiteria di power failure
J6	2	Seleziona collegamento batteria di bordo BT1
J7	2	Seleziona collegamento circuiteria di watch dog
J8	2	Seleziona collegamento circuiteria di terminazione a linea CAN
J9	3	Seleziona la direzione ed il modo operativo per la linea seriale B in RS 422, RS 485
J10 , J11	2	Collegano circuiteria di terminazione e forzatura alla linea seriale B in RS 422, RS 485
JS3	3	Seleziona il tipo di collegamento per il pin 1 di CN3A
JS4	3	Seleziona il tipo di collegamento per il pin 1 di CN3B
JS24	2	Seleziona collegamento della corazza del connettore CN10

FIGURA 27: TABELLA RIASSUNTIVA JUMPERS

Di seguito é riportata una descrizione tabellare delle possibili connessioni dei 14 jumpers con la loro relativa funzione. Per riconoscere tali connessioni sulla scheda si faccia riferimento alla serigrafia della stessa o alle figure 25, 26 di questo manuale, dove viene riportata la numerazione dei pin dei jumpers, che coincide con quella utilizzata nella seguente descrizione. Per l'individuazione dei jumpers a bordo della scheda, si utilizzino invece le figure 29 e 30.
In tutte le seguenti tabelle l'***** indica la connessione di default, ovvero quella impostata in fase di collaudo, con cui la scheda viene fornita.

JUMPERS A 2 VIE

JUMPER	CONNESSIONE	UTILIZZO	DEF.
J1	non connesso	Setta ingresso di configurazione al livello logico 1 (modalità RUN)	*
	connesso	Setta ingresso di configurazione al livello logico 0 (modalità DEBUG)	
J2	non connesso	Non collega il pin 26 di CN1 ai +5 Vdc di alimentazione della scheda	
	connesso	Collega il pin 26 di CN1 ai +5 Vdc di alimentazione della scheda	*
J6	non connesso	Non collega batteria di bordo BT1 alla circuiteria di back up	*
	connesso	Collega batteria di bordo BT1 alla circuiteria di back up	
J7	non connesso	Non collega circuiteria di watch dog alla circuiteria di reset	*
	connesso	Collega circuiteria di watch dog alla circuiteria di reset	
J8	non connesso	Non collega la resistenza di terminazione da 120 Ω alla linea CAN	*
	connesso	Collega la resistenza di terminazione da 120 Ω alla linea CAN	
J10, J11	non connessi	Non collegano la circuiteria di terminazione e forzatura al ricevitore/trasmettitore RS 485 od al ricevitore RS 422, della linea seriale B	*
	connessi	Collegano la circuiteria di terminazione e forzatura al ricevitore/trasmettitore RS 485 od al ricevitore RS 422, della linea seriale B	
JS24	non connesso	Non collega la carcassa metallica del connetore CN10 alla massa galvanicamente isolata CAN GND della linea CAN	*
	connesso	Collega la carcassa metallica del connetore CN10 alla massa galvanicamente isolata CAN GND della linea CAN	

FIGURA 28: TABELLA JUMPERS A 2 VIE

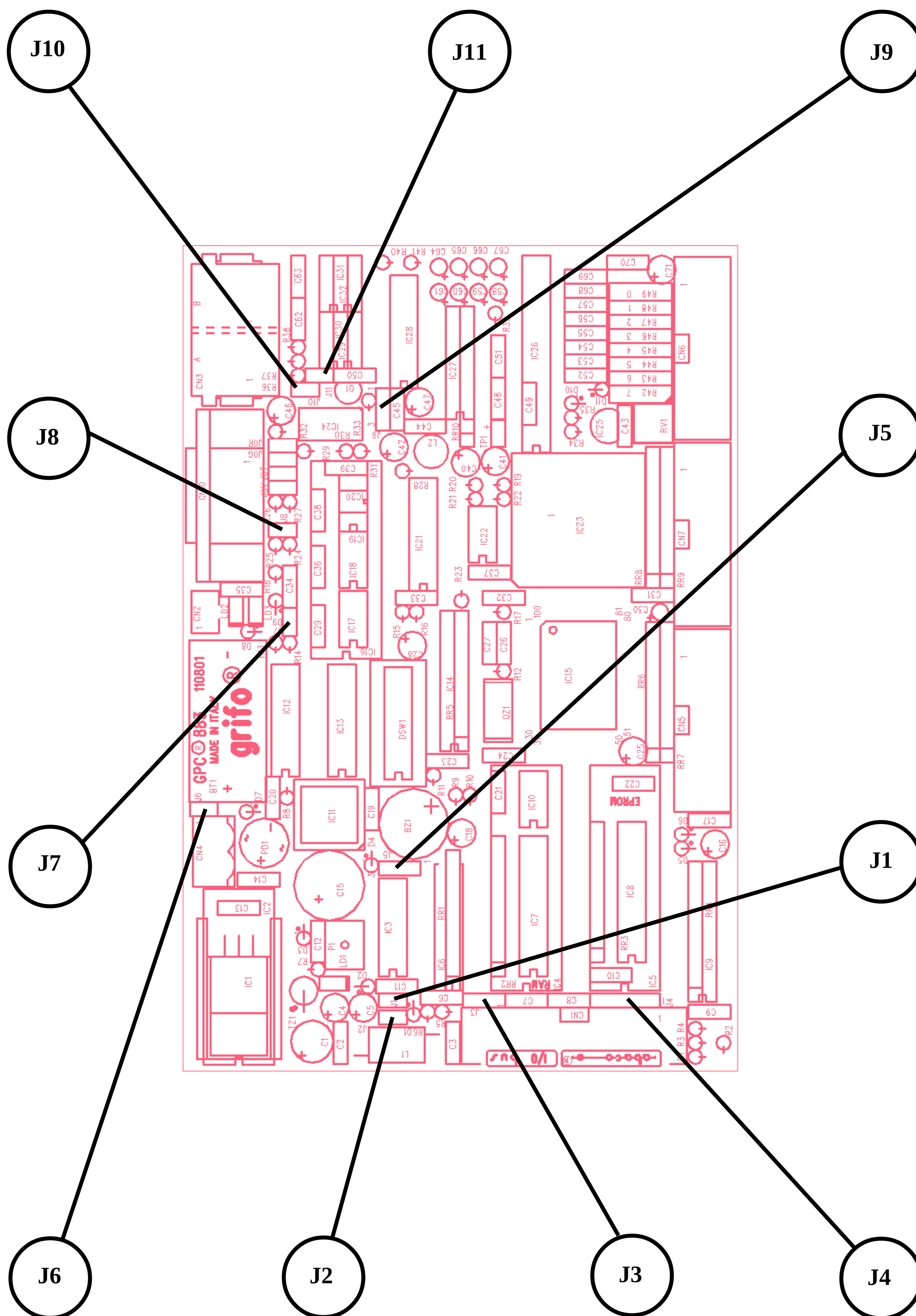


FIGURA 29: DISPOSIZIONE JUMPERS LATO COMPONENTI

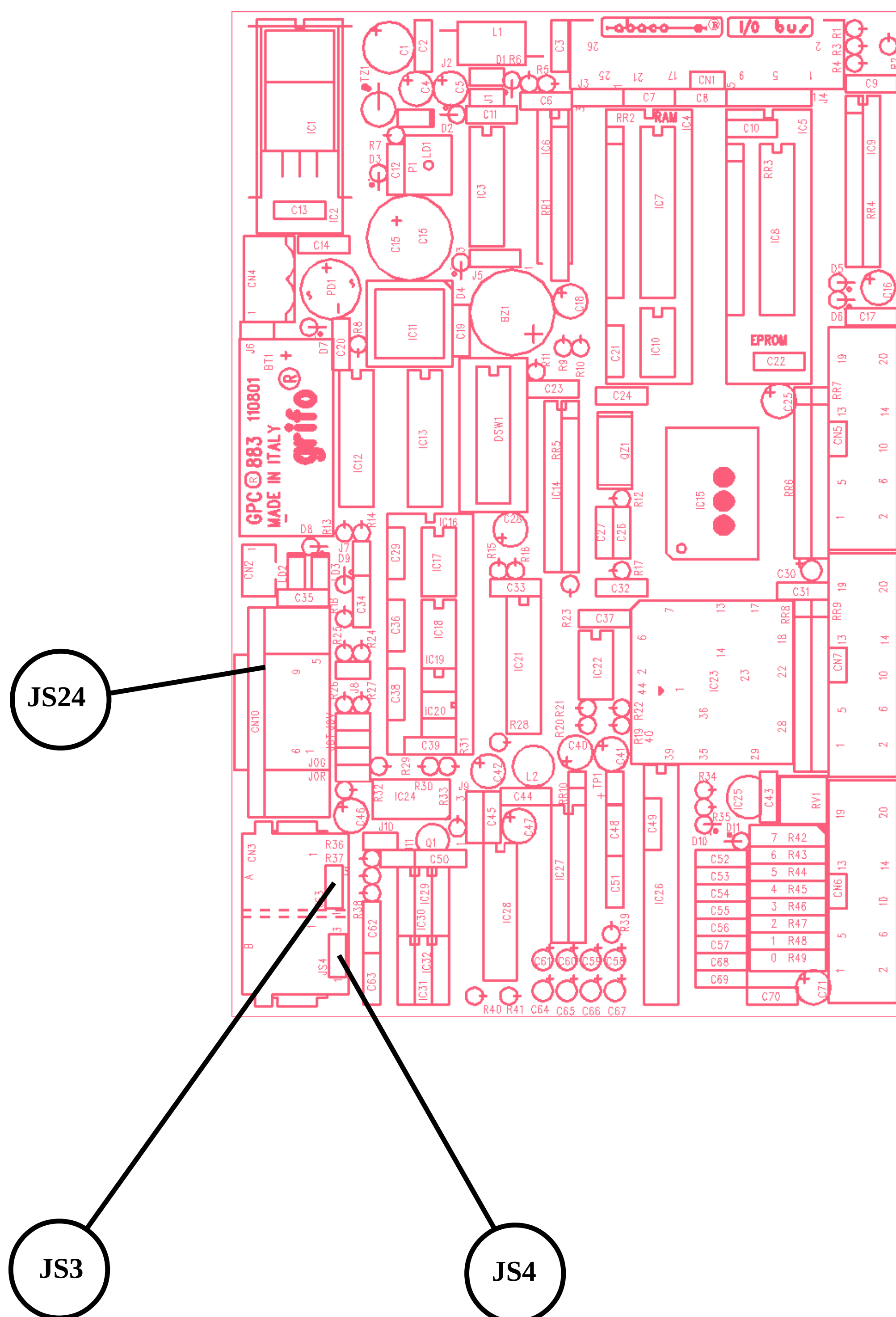


FIGURA 30: DISPOSIZIONE JUMPERS LATO STAGNATURE

JUMPERS A 3 VIE

JUMPER	CONNESSIONE	UTILIZZO	DEF.
J3	posizione 1-2	Predispone zoccolo IC 4 per SRAM da 128K Byte	*
	posizione 2-3	Predispone zoccolo IC 4 per SRAM da 512K Byte	
J5	non connesso	Non collega circuiteria di power failure	*
	posizione 1-2	Collega circuiteria di power failure al segnale di interrupt NMI della CPU	
	posizione 2-3	Collega circuiteria di power failure al segnale di interrupt INT0 della CPU	
J9	posizione 1-2	Seleziona comunicazione seriale sulla linea seriale B in RS 485 (half duplex a 2 fili)	
	posizione 2-3	Seleziona comunicazione seriale sulla linea seriale B in RS 422 (full duplex o half duplex a 4 fili)	*
JS3	posizione 1-2	Collega pin 1 di CN3A a GND	*
	posizione 2-3	Collega pin 1 di CN3A a +5 Vdc	
JS4	posizione 1-2	Collega pin 1 di CN3B a GND	*
	posizione 2-3	Collega pin 1 di CN3B a +5 Vdc	

FIGURA 31: TABELLA JUMPERS A 3 VIE

JUMPERS A 5 VIE

JUMPER	CONNESSIONE	UTILIZZO	DEF.
J4	posizione 1-2 e 3-4	Predispone zoccolo IC5 per EPROM	
	posizione 2-3 e 4-5	Predispone zoccolo IC5 per FLASH EPROM	*

FIGURA 32: TABELLA JUMPERS A 5 VIE

RESET E WATCH DOG

La scheda **GPC® 883** è dotata di una circuiteria di watch dog molto efficiente e di facile gestione software. In particolare le caratteristiche di questa circuiteria sono le seguenti:

- funzionamento astabile;
- tempo d’intervento di circa 1420 msec;
- attivazione via hardware;
- retrigger via software.

Si ricorda che nel funzionamento astabile una volta scaduto il tempo d’intervento la circuiteria si attiva, rimane attiva per il tempo di reset (circa 200 msec) e quindi si disattiva autonomamente. Con il jumper J7 si seleziona se collegare la circuiteria di watch dog esterna alla circuiteria di reset, ovvero si attiva, via hardware, la sua gestione. Il LED LD2 invece segnala l'intervento della circuiteria in modo da poterlo riconoscere visivamente. Per quanto riguarda l’operazione di retrigger si faccia riferimento all'ononimo paragrafo.
In corrispondenza dell'attivazione e sucessiva disattivazione del segnale di /RESET la scheda riprende l’esecuzione del programma salvato su IC5 (EPROM o FLASH EPROM) all'indirizzo FFFF0H della CPU, partendo da una condizione di azzeramento generale.
Si ricorda inoltre che il segnale di /RESET generato dalla scheda é collegato al pin 20 del connettore CN1 e che tra le sorgenti di reset della **GPC® 883**, oltre all'eventuale circuiteria di watch dog, sono sempre presenti la circuiteria di power good, power on ed il pulsante di reset.
Il pulsante di reset, denominato P1, consente di attivare la circuiteria di /RESET con le dovute modalità e tempistiche; la sua funzione principale é quella di uscire da condizioni di loop infinito, soprattutto durante la fase di debug oppure per rieseguire il programma applicativo senza dover interrompere l'alimentazione della scheda. Per una facile individuazione di tale pulsante a bordo scheda, si faccia riferimento alla figura 24.

SEGNALAZIONI VISIVE

La scheda **GPC® 883** é dotata delle segnalazioni visive descritte nella seguente tabella:

LED	COLORE	FUNZIONE
LD1	Rosso	Visualizza presenza della tensione di alimentazione a +5 Vdc
LD2	Rosso	Visualizza attivazione della circuiteria di watch dog
LD3	Verde	LED di attività gestito via software

FIGURA 33: TABELLA DELLE SEGNALAZIONI VISIVE

La funzione principale di questi LEDs é quella di fornire un'indicazione visiva dello stato della scheda, facilitando quindi le operazioni di debug e di verifica di funzionamento di tutto il sistema. Per una più facile individuazione di tali segnalazioni visive, si faccia riferimento alla figura 24, mentre per ulteriori informazioni sull'attivazione dei LED si faccia riferimento ai paragrafi "ALIMENTAZIONE", "RESET E WATCH DOG" e "LED DI ATTIVITA'".

INTERRUPTS

Una caratteristica peculiare della **GPC® 883** è la notevole potenza nella gestione delle interruzioni. Di seguito viene riportata una breve descrizione di quali sono i dispositivi che possono generare interrupts e con quale modalità; per quanto riguarda la gestione di tali interrupts si faccia riferimento ai data sheets del microprocessore oppure all'appendice B di questo manuale.

- **ABACO®** I/O BUS	->	Genera un NMI sulla CPU, tramite la linea /NMI BUS di CN1. Genera un INT0 sulla CPU, tramite la linea /INT BUS di CN1.
- Power failure	->	Genera un NMI o INT0 sulla CPU, a seconda del collegamento di J5.
- Real Time Clock	->	Genera un INT1 sulla CPU.
- A/D converter	->	Genera un INT3 sulla CPU.
- PIO	->	Genera un INT5 sulla CPU, tramite linea PIO PD.2 di CN5. Genera un INT6 sulla CPU, tramite linea PIO PD.3 di CN5.
- Controllore CAN	->	Genera un INT4 sulla CPU.
- Periferiche della CPU	->	Generano un interrupt interno. In particolare le possibili sorgenti d'interrupt interno sono le sezioni: TC 0, TC 1, TC 2, DMA 0, DMA 1, ASP 0,ASP 1,interrupt software.

Sulla scheda é presente un gestore d'interrupt (ICU) che consente di attivare, disattivare, mascherare le sorgenti d'interrupt e che regolamenta l'attivazione contemporanea di più interrupts. In questo modo l'utente ha sempre la possibilità di rispondere in maniera efficace e veloce a qualsiasi evento esterno, stabilendo anche la priorità delle varie sorgenti.

ALIMENTAZIONE

La scheda **GPC® 883** dispone di una efficiente circuiteria che si presta a risolvere in modo comodo ed efficace il problema dell'alimentazione della scheda in qualsiasi condizione di utilizzo. Di seguito vengono riportate le possibili configurazioni della sezione alimentatrice:

- Senza sezione alimentatrice (configurazione di default):
In questa configurazione la scheda deve essere alimentata da una tensione di +5 Vdc che può essere fornita sugli appositi pin di CN1 o CN3A,CN3B. In caso di alimentazione fornita tramite i connettori plug CN3A o CN3B (ad esempio in sistemi distribuiti in cui più schede sono collegate in rete, tramite un solo cavo che porta sia i segnali di comunicazione che quelli di alimentazione), si devono configurare opportunamente i jumpers JS3 e JS4.

*- Con sezione alimentatrice lineare (opzione **.ALIM12**):*
In questa configurazione la scheda deve essere alimentata da una tensione di 6÷12 Vac, o corrispondente tensione continua 9÷16 Vdc, che deve essere fornita sui pin 1 e 2 di CN4.

*- Con sezione alimentatrice switching (opzione **.SW**):*
In questa configurazione la scheda deve essere alimentata da una tensione di 8÷24 Vac, o corrispondente tensione continua 12÷34 Vdc, che deve essere fornita sui pin 1 e 2 di CN4.

Indipendentemente dalla sezione alimentatrice scelta la **GPC® 883** é sempre dotata di un'efficace circuiteria di protezione che si preoccupa di proteggere la scheda da sovratensioni o dal rumore del campo. Si ricorda che la selezione del tipo di sezione alimentatrice della scheda, deve avvenire in fase di ordine della stessa; infatti questa scelta implica una diversa configurazione hardware, che deve essere effettuata dal personale addetto.

Il LED LD1 visulizza la presenza della tensione di alimentazione e deve quindi risultare sempre acceso in qualsiasi configurazione e condizione operativa. La sua disattivazione segnala immediatamente che la scheda non funziona a causa dell'assenza alimentazione.

Il jumper J1 collega il positivo della tensione di alimentazione della scheda al connettore dell'**ABACO®** I/O BUS CN1 e deve quindi essere:
- non connesso quando si collega la scheda provvista di sezione alimentatrice ad un sistema a sua volta provvisto di propria sezione alimentatrice, per evitare conflitti tra due alimentazioni diverse;
- connesso quando la **GPC® 883** non ha sezione alimentatrice (in questo caso l'alimentazione fornita su CN1 viene usata per alimentare tutta la scheda) oppure quando solo la **GPC® 883** ha la sezione alimentatrice (in questo caso le schede periferiche collegate su CN1 sono alimentate dalla stessa **GPC® 883**.

Quando la scheda é provvista di sezione alimentatrice e viene alimentata da una tensione continua su CN4, il segnale di massa di questa alimentazione non coincide con la linea GND presente sui vari connettori. In dettaglio le due masse saranno ad una differenza di potenziale di circa 0,6 V caratteristico del ponte raddrizzatore d'ingresso.

Si ricorda che la tensione di alimentazione della logica di bordo (segnali +5 Vdc e GND) é galvanicamente isolata dalla tensione di alimentazione dell'interfaccia CAN. Quindi il segnale GND, presente sui connettori della scheda non deve essere collegato al segnale CAN GND presente su CN10. Nel caso di particolari esigenze dell'utente é possibile avere la **GPC® 883** con la linea CAN non isolata; per ulteriori informazioni contattare direttamente la **grifo®**.

La sezione di A/D converter utilizza la tensione di alimentazione della scheda opportunamente filtrata e distribuita. Per ragioni di schermatura e disposizione piste, la massa di tale sezione é stata chiamata AGND in modo da distinguerla da quella di alimentazione GND, anche se le due sono elettricamente connesse.

Al fine di ridurre i consumi della scheda si può utilizzare la modalità operativa di power save della CPU. Questa modalità consente di definire la frequenza di lavoro della CPU e può essere selezionata programmando l'apposito registro SYSCON, interno al microprocessore. Il programma applicativo sviluppato dall'utente può quindi ridurre il consumo sull'alimentazione fino ad un minimo di 120 mA ed eventualmente ripristinare il funzionamento normale in corrispondenza di un evento presatbilito come ad esempio un interrupt, variazione di un ingresso digitale e/o analogico, intervallo di tempo trascorso, ecc.

Per ulteriori informazioni si faccia riferimento al paragrafo “CARATTERISTICHE ELETTRICHE”.

SELEZIONE COMUNICAZIONE SERIALE

La linea di comunicazione seriale A della scheda **GPC® 883** può essere bufferata solo in RS 232, mentre la linea seriale B può essere bufferata in RS 232, RS 422, RS 485 o current loop. Dal punto di vista software su entrambe le linee può essere definito il protocollo fisico di comunicazione tramite la programmazione di alcuni registri interni della CPU. In dettaglio possono essere programmate per lavorare con 7,8,9 bit per carattere; parità pari, dispari o nessuna; 1 bit di stop; con baud rate standard e non standard, fino a 115200 Baud.
La selezione del protocollo elettrico della seriale B avviene via hardware e viene effettuata tramite un'opportuna configurazione dei jumpers di bordo, come descritto nelle precedenti tabelle, e l'installazione di adeguati driver di comunicazione. Alcuni componenti necessari per le configurazioni RS 422, RS 485 e current loop non sono montati e collaudati sulla scheda in configurazione di default; per questo la prima configurazione della seriale B non in RS 232 deve essere sempre effettuata dai tecnici **grifo®**. A questo punto l'utente può cambiare autonomamente la configurazione seguendo le informazioni sotto riportate:

- LINEA SERIALE B=ASP0 SETTATA IN RS 232 (configurazione default)

			IC28	= driver MAX 202
J9	=	indifferente	IC29	= nessun componente
J10, J11	=	non connessi	IC30	= nessun componente
			IC31	= nessun componente
			IC32	= nessun componente

- LINEA SERIALE B=ASP0 SETTATA IN CURRENT LOOP (opzione .CLOOP)

			IC28	= nessun componente
J9	=	indifferente	IC29	= nessun componente
J10, J11	=	non connessi	IC30	= driver HP 4100
			IC31	= driver HP 4200
			IC32	= nessun componente

Da ricordare che l'interfaccia seriale in current loop é di tipo passivo e si deve quindi collegare una linea current loop attiva, ovvero provvista di un proprio alimentatore come descritto nelle figure 12÷14. L'interfaccia current loop può essere utilizzata per realizzare sia connessioni punto punto che reti multipunto con un collegamento a 4 o 2 fili.

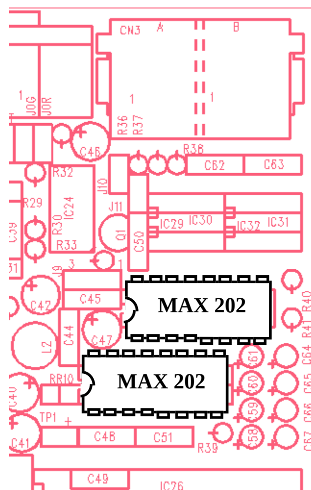
- LINEA SERIALE B=ASP0 SETTATA IN RS 422 (opzione .RS 422)

			IC28	= nessun componente
J9	=	posizione 2-3	IC29	= driver SN 75176 o MAX 483
J10, J11	=	(*)	IC30	= nessun componente
			IC31	= nessun componente
			IC32	= driver SN 75176 o MAX 483

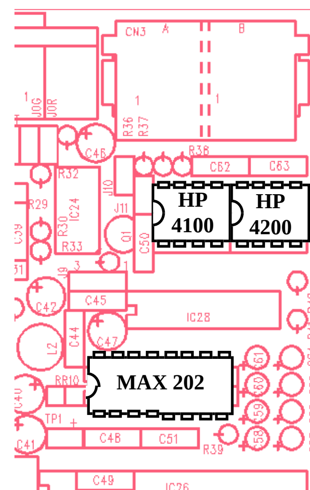
Lo stato del segnale /RTS0=PIO20 (gestito via software con i registri dell'ASP0 o PIO) consente di abilitare o disabilitare il trasmettitore come segue:

/RTS0=PIO20 = livello basso	= stato logico 0	->	trasmettitore attivo
/RTS0=PIO20 = livello alto	= stato logico 1	->	trasmettitore disattivo

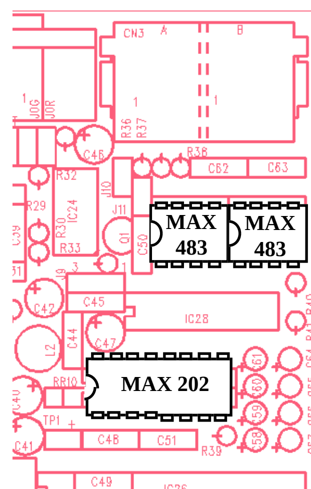
Per sistemi punto punto, la linea /RTS0=PIO20 può essere mantenuta sempre bassa (trasmettitore sempre attivo), mentre per reti multipunto si deve attivare il trasmettitore solo in corrispondenza della trasmissione. La comunicazione RS 422 é di tipo full duplex.



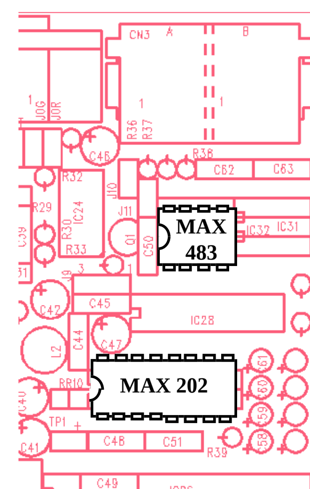
Seriale B = ASP0 in RS 232



Seriale B = ASP0 in current loop



Seriale B = ASP0 in RS 422



Seriale B = ASP0 in RS 485

FIGURA 34: DISPOSIZIONE DRIVER PER COMUNICAZIONE SERIALE

- LINEA SERIALE B=ASP0 SETTATA IN RS 485 (opzione .RS 485)

			IC28	= nessun componente
J9	=	posizione 1-2	IC29	= driver SN 75176 o MAX 483
J10, J11	=	(*)	IC30	= nessun componente
			IC31	= nessun componente
			IC32	= nessun componente

In questa modalità le linee da utilizzare sono i pin 4 e 5 di CN3B, che quindi diventano le linee di trasmissione o ricezione a seconda dello stato del segnale /RTS0=PIO20 (gestito via software con i registri dell'ASP0 o PIO) come segue:

/RTS0=PIO20 = livello basso = stato logico 0 -> linea in trasmissione
/RTS0=PIO20 = livello alto = stato logico 1 -> linea in ricezione

Questa comunicazione la si utilizza sia per connessioni punto punto che multipunto con una comunicazione half duplex. Sempre in questa modalità si riceve quanto trasmesso, in modo da fornire al sistema la possibilità di verificare autonomamente la riuscita della trasmissione; infatti in caso di conflitti sulla linea, quanto trasmesso non viene ricevuto correttamente e viceversa.

(*) Nel caso si utilizzi la linea seriale in RS 422 o RS 485, con i jumpers J10 e J11 é possibile connettere la circuiteria di terminazione e forzatura sulla linea . Tale circuiteria deve essere sempre presente nel caso di sistemi punto punto, mentre nel caso di sistemi multipunto, deve essere collegata solo sulle schede che risultano essere alla maggior distanza, ovvero ai capi della linea di comunicazione.
In fase di reset o power on, il segnale /RTS0=PIO20 è mantenuto a livello logico alto di conseguenza in seguito ad una di queste fasi il driver RS 485 è in ricezione o il driver di trasmissione RS 422 è disattivo, in modo da eliminare eventuali conflittualità sulla linea di comunicazione.

Si ricorda che i segnali di handshake (RTS e CTS) disponibili con le linee seriali bufferate in RS 232 ed utilizzati per le linee bufferate in RS 422, RS 485 possono essere gestiti sia tramite gli ononimi segnali delle sezioni ASP0 e ASP1 che tramite le linee del PIO. In particolare vale la seguente corrispondenza:

RTSA RS232	->	/RTS1	=	PIO19
CTSA RS232	->	/CTS1	=	PIO18
RTSB RS232	->	/RTS0	=	PIO20
CTSB RS232	->	/CTS0	=	PIO21

Per ulteriori informazioni relative alla comunicazione seriale fare riferimento agli esempi di collegamento delle figure 8÷14 ed al paragrafo "PIN MULTIFUNZIONE".

JUMPERS A STAGNO

La connessione di default dei jumpers a stagno denominati **JSxx**, é effettuata con una sottile pista sul lato stagnature. Quindi, se tale configurazione deve essere variata, si deve prima tagliare la pista con un taglierino affilato e poi effettuare la connessione richiesta con uno stagnatore di bassa potenza, utilizzando dello stagno non corrosivo.

SELEZIONE MEMORIE

La **GPC® 883** può montare fino ad un massimo di 1032 Kbytes di memoria variamente suddivisa. In particolare per la configurazione seguire le informazioni riportate nella seguente tabella:

IC	DISPOSITIVO	DIMENSIONE	CONFIGURAZIONE JUMPER
5	EPROM	128K Byte	J4 in posizione 1-2 e 3-4
	EPROM	256K Byte	J4 in posizione 1-2 e 3-4
	EPROM	512K Byte	J4 in posizione 1-2 e 3-4
	FLASH EPROM	128K Byte	J4 in posizione 2-3 e 4-5
	FLASH EPROM	512K Byte	J4 in posizione 2-3 e 4-5
4	RAM	128K Byte	J3 in posizione 1-2
	RAM	512K Byte	J3 in posizione 2-3
22	EEPROM	256÷8K Byte	-

FIGURA 35: TABELLA DI SELEZIONE MEMORIE

Tutti i dispositivi sopra descritti devono essere con pin out di tipo JEDEC a parte l'EEPROM seriale di IC 22 che deve essere richiesta alla **grifo®** in fase di ordine della scheda. Per quanto riguarda le sigle dei vari dispositivi che possono essere montati, fare riferimento alla documentazione della casa costruttrice.
Normalmente la **GPC® 883** é fornita nella sua configurazione di default con 128K SRAM su IC4 e 512 byte di EEPROM seriale su IC22; ogni configurazione diversa può essere autonomamente montata dall'utente oppure richiesta nella fase di ordine. Sotto sono riportate i codici delle opzioni di memoria disponibili:

.512K	->	512K Byte di SRAM su IC4
.EE08	->	1K Byte EEPROM seriale su IC22
.EE16	->	2K Byte EEPROM seriale su IC22
.EE64	->	8K Byte EEPROM seriale su IC22

Per ulteriori informazioni e costi delle opzioni, contattare direttamente la **grifo®**, mentre per una facile individuazione dei dispositivi di memoria fare riferimento alla figura 24.

BACK UP

La **GPC® 883** é provvista di una batteria al litio BT1 che provvede a tamponare la SRAM ed il Real Time Clock di bordo anche in assenza della tensione di alimentazione. Il jumper J6 provvede a collegare o meno questa batteria in modo da salvaguardarne la durata prima dell'installazione o in tutti i casi in cui il back up non é necessario. Una seconda batteria esterna, con tensione superiore a

quella di bordo, può essere collegata alla circuiteria di back up tramite il connettore CN2: quest'ultima non é interessata dalla configurazione del jumper J6 e sostituisce a tutti gli effetti la BT1. Per la scelta della batteria esterna di back up seguire le indicazioni del paragrafo "CARATTERISTICHE ELETTRICHE", per la sua individuazione si veda la figura 24 mentre per acquisire lo stato di carica della/delle batterie di back up si veda l'apposito paragrafo "STATO DELLA BATTERIA".

POWER FAILURE

In abbinamento alla circuiteria di power management gestita dalla CPU della **GPC® 883** é inoltre disponibile un'interessante circuiteria di power failure. Quest'ultima, con il jumper J5, può essere collegata a due diversi interrupt (NMI o INT0).
La circuiteria si preoccupa di controllare la tensione di alimentazione e quando questa scende al valore di soglia (**52 mV** prima dell'intervento del reset), provvede ad attivare l'uscita richiedendo l'attenzione della CPU nel caso che J5 sia collegato in una delle due posizioni.
Da notare che il tempo che intercorre tra l'attivazione del power failure e quello del reset, varia in funzione del tipo di alimentazione della scheda; questo normalmente é nell'ordine dei 100 μsec, sufficienti solo per eseguire procedure di risposta veloci (ad esempio il salvataggio di un flag nella memoria tamponata).
L'uso classico di questa circuiteria é quello di informare la scheda dell'imminente caduta della tensione di alimentazione, in modo da salvare le condizioni di stato necessarie, grazie alla carica residua della sezione alimentatrice.

COLLEGAMENTO LINEA CAN

Il jumper J8 ha il compito di collegare o meno l'apposita resistenza di terminazione della linea CAN come descritto nella tabella di figura 28. Il CAN BUS deve fisicamente coincidere con una linea differenziale con impedenza di 60 Ω e per questo le resistenze di terminazione devono essere collegate in modo da ricreare questa impedenza. In particolare tale collegamento deve essere sempre effettuato in caso di sistemi punto punto, mentre nel caso di sistemi multipunto, deve essere collegata solo sulle schede che risultano essere alla maggior distanza, ovvero ai capi della linea di comunicazione CAN (vedere esempio di figura 23).
La corretta terminazione della linea CAN contribuisce notevolmente al funzionamento della comunicazione, infatti l'interfaccia di linea della **GPC® 883** é in grado di sopprimere i transienti e di essere immune ai disturbi di radio frequenza ed elettromagnetici, solo se il collegamento con il campo é effettuato correttamente.
La scheda é inoltre provvista di una circuiteria di schermatura che collega dinamicamente, ma non staticamente, la carcassa del connettore CAN CN10 alla massa di alimentazione isolata CAN GND.
Il jumper a stagno JS24 ha il compito di collegare direttamente la carcassa di CN10 al segnale CAN GND ottenendo quindi un collegamento statico. Con tale circuiteria, quando si utilizza del cavo schermato per la linea CAN, é possibile schermare anche il collegamento fisico alla scheda ottenendo la massima protezione contro i disturbi esterni.

PIN MULTIFUNZIONE

Il microcontrollore Am188ES usato a bordo della **GPC® 883** é provvisto di due port a 16 bit di I/O generico per un totale di 32 linee. Molte di queste linee sono fisicamente multiplexate all'interno dello stesso microprocessore e possono quindi assumere funzionalità diverse a seconda della programmazione software effettuata. Nella seguente tabella viene riportato l'elenco dei segnali a disposizione dell'utente con la relativa descrizione di tutte le possibili funzioni e del loro uso sulla scheda.

SEGNALE PIO	FUNZIONI MPX	USO SU GPC® 883
PIO 00	PIO 00 ; TMR IN 1	PD.0 su CN5
PIO 01	PIO 01 ; TMR OUT 1	IO0 su CN5
PIO 04	PIO 04 ; DTR	SDA EEPROM seriale
PIO 06	PIO 06 ; SRDY	LED attività
PIO 10	PIO 10 ; TMR OUT 0	IO1 su CN5
PIO 11	PIO 11 ; TMR IN 0	PD.1 su CN5
PIO 12	PIO 12 ; DRQ 0 ; INT5	PD.2 su CN5
PIO 13	PIO 13 ; DRQ 1 ; INT6	PD.3 su CN5
PIO 14	PIO 14	PD.4 su CN5
PIO 15	PIO 15	PD.5 su CN5
PIO 18	PIO 18 ; /CTS1 ; /ENRX1	/CTSA per CN3A
PIO 19	PIO 19 ; /RTS1 ; /RTR1	/RTSA per CN3A
PIO 20	PIO 20 ; /RTS0 ; /RTR0	/CTSB per CN3B
PIO 21	PIO 21 ; /CTS0 ; /ENRX0	/RTSB per CN3B ; gestore RS 422,RS 485
PIO 24	PIO 24	PD.6 su CN5
PIO 25	PIO 25	Buzzer
PIO 26	PIO 26	Acquisizione stato carica batterie
PIO 29	PIO 29 ; /CLKDIV2	SCL EEPROM seriale
PIO 31	PIO 31 ; PWD ; INT2	PD.7 su CN5

FIGURA 36: TABELLA PIN MULTIFUNZIONE

L'utente può arbitrariamente decidere il tipo di funzionalità dei segnali PIO collegati al connettore CN5, mentre deve programmare con scrupolosa attenzione i segnali dedicati alla gestione dell'EEPROM seriale e degli handshake. Normalmente i pacchetti software sviluppati per la scheda si preoccupano già di effettuare il giusto settaggio di tutte le linee del PIO ed all'utente rimane solo da modificare quelle riportate in tabella facendo attenzione a non alterare le rimanenti. Una variaziane nell'inizializzazione o nello stato delle linee PIO non descritte in tabella, può causare un malfunzionamento od un blocco nell'esecuzione del programma applicativo.

La seguente tabella riporta lo stato delle linee di I/O, presenti sul connettore CN5, durante la fase di power on o reset.

PIN CN5	STATO DURANTE POWER ON O RESET
PD.0	Input con pull-up interno
PD.1	Input con pull-up interno
PD.2	Input con pull-up interno
PD.3	Input con pull-up interno
PD.4	Input con pull-up interno
PD.5	Input con pull-up interno
PD.6	Input con pull-up interno
PD.7	Input con pull-up interno
IO0	Input con pull-down interno
IO1	Input con pull-down interno

FIGURA 37: STATO DEI PIN DI CN5 DURANTE POWER ON O RESET

Il valore della resisten za interna di pull up o pull down é di circa 10 KΩ, quindi in assenza di carichi esterni a bassa impedenza, le relative linee su CN5, durante la fase di power on o reset, si settano rispettivamente allo stato logico 1 o 0. La resistenza di pull up a bordo scheda (vedere figura 16) contribuisce a rafforzare il pull up interno e crea un partitore con il pull down interno: i segnali IO0 e IO1 si troveranno quindi ad uno stato intermedio di circa 2,5 V durante le fasi di reset e power on.



FIGURA 38: FOTO SCHEDA DALL'ALTO

DESCRIZIONE SOFTWARE

Questa scheda ha la possibilità di usufruire di una ricca serie di strutture software che consentono di utilizzarne il modulo come sistema di sviluppo di sé stesso, sia in assembler che in linguaggi ad alto livello. In questo modo l'utente può facilmente sviluppare le applicazioni necessarie in un tempo veramente corto. In generale la scheda può utilizzare tutte le risorse software disponibili per il processore montato, ovvero i numerosi pacchetti ideati per l'8086 e derivati. Si ricorda inoltre che anche tutto il software disponibili sui P.C. potrebbe essere eseguito sulla **GPC® 883** ma che realmente ciò non é possibile a causa della sua incompatibilità hardware con i P.C.
Tutti i pacchetti di sviluppo software forniti dalla **grifo®** sono sempre accompagnati da esempi che illustrano come gestire ogni sezione della scheda e da una completa documentazione d'uso.
Tra questi ricordiamo:

GET188
Completo programma di EDITOR , Comunicazione e gestione delle memorie di massa per le schede della famiglia 188. Questo programma, sviluppato dalla **grifo®**, consente di operare in condizioni ottimali, tutte le volte che si deve usare il GDOS o la versione per FLASH EPROM FGDOS. Viene fornito in abbinamento all'aquisto di uno dei pacchetti citati e viene personalizzato con il nome ed i dati dell'acquirente. Una serie di comodi menù a tendina facilita l'uso del programma, il quale può funzionare anche in abbinamento ad un mouse. Il programma, oltre che girare in ambiente MS-DOS e WINDOWS, gira tranquillamente anche sulle macchine MACINTOSH in abbinamento al programma SOFT-PC. Viene fornito su dischetti MS-DOS da 3"1/2 con relativa documentazione sul manuale GDOS 188.

GDOS 883
Tools di sviluppo completo per le schede della fam. 80. Viene fornito in abbinamento al programma GET188, per consentire un immediato e pieno utilizzo di questo potente strumento di sviluppo. Il GDOS può essere concettualmente diviso in due distinte strutture. Una struttura lavora essenzialmente su PC, mantenendo il collegamento con la seconda tramite la linea seriale. La seconda risiede in EPROM ed opera a bordo scheda. La parte a bordo scheda é essenzialmente un potente sistema operativo che si preoccupa di eseguire tutte quelle funzioni a più basso livello e nello stesso tempo consente di poter operare con alcuni linguaggi ad alto livello direttamente a bordo scheda. L'abbinamento delle due strutture fa si che la scheda ed il PC si comportino come un'unica macchina. Infatti la scheda usa, come se fossero le proprie, le risorse del PC come le memoria di massa quali i floppy disk, l'hard disk; la stampante ecc. Il tutto avviene in modo completamente trasparente per l'utente il quale usa questo tipo di macchina virtuale esattamente come é abituato ad adoperare il suo PC.
Il GDOS, oltre ai tipici drivers del PC, gestisce come RAM disk una porzione della memoria RAM di bordo. Questo significa che il dispositivo di RAM a bordo scheda, tamponato tramite batteria, può essere gestito in modo diretto dai linguaggi ad alto livello, trattando comodamente come files, le informazioni da archiviare o ricercare.
Nel GDOS 883 é inoltre presente un potente monitor debugger che é in grado di caricare, eseguire e debuggare programmi applicativi generati da un generico cross assembler o compilatore esterno. Il pacchetto viene fornito in EPROM, in abbinamento al disco e CD, utility , alcuni esempi di uso e la relativa manualistica sul istema operativo.

FGDOS 883
Caratteristiche analoghe al GDOS, con la differenza che é in grado di programmare e cancellare le FLASH EPROM a bordo scheda, con i programmi generati dall'utente. In questo modo non é

necessario un programmatore di EPROM esterno per congelare il programma. E' inoltre possibile, tramite un PC portatile, intervenire direttamente a bordo macchina per cambiare il programma di gestione.
l pacchetto viene fornito in FLASH EPROM, in abbinamento al disco e CD, utility , alcuni esempi di uso e la relativa manualistica sul istema operativo.

PASCAL 86
Completo e molto efficiente Compilatore PASCAL in grado di generare un codice romato per la famiglia 86 di CPU. Ha delle caratteristiche operative analoghe a quelle del Turbo PASCAL Ver.3 della Borland, a cui si fà riferimento per quanto riguarda sia le caratteristiche che la manualistica. Il PASCAL 86 lavora in abbinamento ad una delle varie versioni di sistema operativo GDOS. Le modalità di emulazione terminale offerta dal programma GET188, supportano pienamente il tipico editor a pieno schermo del PASCAL, compresa la gestione degli attributi. Sfruttando la possibilità di gestione di RAM disk offerta dal GDOS, si possono sfruttare appieno le istruzioni di file system offerte dal linguaggio. Genera un codice molto compatto, il quale risiede e viene eseguito in EPROM o FLASH EPROM e che richiede pochissimo spazio di RAM.
Il programma viene fornito in EPROM o FLASH EPROM, assieme al sistema operativo GDOS, e su dischetto MS-DOS in abbinamento alle note tecniche e ad una serie di esempi.

GCTR 883
Completo pacchetto di sviluppo, per la scheda **GPC® 883**, che consente di sviluppare dei programmi applicativi adoperando un Compilatore C della Borland, in abbinamento ad un normale PC. Un potentissimo debugger simbolico remoto, residente sulla scheda, consente di operare in modalità sorgente, con delle prestazioni e delle comodità di uso analoghe a quelle di un emulatore in circuit, grazie ad un semplice collegamento seriale. Alla fine delle operazioni é possibile congelare il programma in EPROM o FLASH EPROM, con un uso molto limitato delle risorse RAM.
Viene fornito in EPROM o FLASH EPROM, una serie di floppy disk e CD con la relativa manualistica tecnica e gli esempi d'uso.

HTC 86
Cross compilatore C professionale della Hi-Tech Software. Questo compilatore é estremamente veloce e genera pochissimo codice. Questo risultato é ottenuto grazie a delle avanzate tecniche di ottimizzazione del codice generato, basato su tecniche di Intelligenza artificiale che gli consentono di ottenere un codice compatto ed estremamente veloce. Il pacchetto comprende IDE, Compilatore, Ottimizzatore del codice, Assemblatore, Linker, Debugger remoto, ecc. Il compilatore segue completamente lo standard ANSI/ISO C ed é provvisto del sorgente di tutte le librerie. Una volta fatto il porting del modulo di debugger remoto, consente di provare il software direttamente nell'hardware in sperimentazione. Questo tipo di specializzazione del debugger remoto é già disponibile, e viene fornito, per tutte le schede di CPU della **grifo®**. Il pacchetto software viene fornito su CD, completo di un esauriente manuale.
Questa versione supporta le CPU 8088, 8086, 80188, 80C188, 80186, 80C186, 80286, V20, V30, V25, ecc.

DDS MICRO C 86
E' un comodo pacchetto software, a basso costo, che tramite un completo IDE permette di utilizzare un editor, un compilatore "C" (integer), un assemblatore, un linker e un debugger remoto abbinato ad un monitor. Sono inclusi i sorgenti delle librerie, una serie di utility ed una ricca documentazione su dischetto da 3"1/2 nel formato MS-DOS.

MAPPAGGI ED INDIRIZZAMENTI

In questo capitolo ci occuperemo di fornire tutte le informazioni sulle caratteristiche hardware della scheda relative al mappaggio delle memorie, delle periferiche, dei registri, ecc. necessarie per la gestione software delle sezioni componenti.

MAPPAGGIO DELLE RISORSE DI BORDO

La gestione delle risorse della scheda è interamente affidata alla sezione di generazione segnali di controllo periferiche (CSU) del microprocessore e alla logica di controllo.
Complessivamente la CPU Am188ES indirizza direttamente 1M Byte di memoria quindi alla logica di controllo è assegnato il compito di allocare lo spazio logico d'indirizzamento delle memorie nello spazio fisico massimo di 1032K Byte. Questa gestione è effettuata dalla sezione CSU del microprocessore che é in grado di generare segnali di chip select in corrispondenza di determinati indirizzi fissati dall'utente, via software tramite la programmazione dei suoi registri interni.
Per quanto riguarda l'I/O la CPU gestisce direttamente 64K Byte indirizzi in cui possono essere allocate le periferiche della scheda. Anche in questo caso la sezione CSU del microprocessore é in grado di generare segnali di chip select in corrispondenza di indirizzi prefissati via software.
Infine alcune periferiche sono comandate direttamente da linee del PIO mantenendo sempre una completa e semplice gestione software.
Con questa prerogativa il mappaggio della scheda é arbitrariamente scelto dalla programmazione della CPU e per l'utente é sufficiente conoscere i collegamenti hardware effettuati a bordo scheda.
Riassumendo l'elenco dei dispositivi mappati ed il relativo collegamento hardware nei confronti dei segnali della CPU é il seguente:

DISPOSITIVO	SEGNALI	INDIRIZZO
EPROM ; FLASH EPROM	/UCS	............H ÷ FFFFFH
RAM	/LCS	00000H ÷ ....H
EEPROM	PIO04 ; PIO29	-
LED ATTIVITA' LD3	PIO 06	-
BUZZER	PIO 25	-
STATO BATTERIE	PIO 26	-
ABACO® I/O BUS	/PCS0	INDPCS0+00H ÷ INDPCS0+FFH
REAL TIME CLOCK	/PCS1	INDPCS1+00H ÷ INDPCS1+0FH
JUMPER J1	/PCS1	INDPCS1
A/D MAX197	/PCS5	INDPCS5+00H ÷ INDPCS5+3FH
PPI 82C55	/PCS5	INDPCS5+40H ÷ INDPCS5+7FH
WATCH DOG	/PCS5	INDPCS5+80H ÷ INDPCS5+BFH
DIP SWITCH DSW1	/PCS5	INDPCS5+C0H ÷ INDPCS5+FFH
CAN CONTROLLER SJA1000	/PCS6	INDPCS6+00H ÷ INDPCS5+7FH

FIGURA 39: COLLEGAMENTO HARDWARE DEI DISPOSITIVI

Nella precedente tabella compaiono alcuni valori non definiti ed alcune indicazioni generiche INDPCSx: tutti questi valori possono essere definiti programmando la sezione CSU. Normalmente i pacchetti software sviluppati per la scheda si preoccupano già di effettuare una programmazione della CSU con modalità descritte nella documentazione d'uso degli stessi pacchetti software.

MAPPAGGIO ABACO® I/O BUS

La CSU della **GPC® 883** provvede anche alla gestione dell'**ABACO®** I/O BUS, definendo gli indirizzi in cui tale BUS viene allocato; come si può notare dallasucessiva figura 39, tale BUS è indirizzato in corrispondenza degli indirizzi del /PCS0 per una estensione di 256 bytes. Un accesso in I/O in un qualsiasi indirizzo compreso in questo range abilita il segnale /IORQ e tutti gli altri segnali di controllo di CN1. Nella fase di mappaggio di eventuali schede periferiche collegate alla scheda, solo il byte meno significativo dell'indirizzo di I/O é significativo in quanto l'**ABACO®** I/O BUS prevede solo 8 bit d'indirizzamento ed 8 bit di dati.

MAPPAGGIO MEMORIE

Sulla scheda i 1032K Byte di memoria che possono essere montati sono così allocati:

Fino a 512K Byte di EPROM o 512K Byte di FLASH EPROM allocati nello spazio di memoria.
Fino a 512K Byte di SRAM allocati nello spazio di memoria.
Fino a 8K Byte di EEPROM seriale allocati indirettamente nello spazio di I/O.

La **GPC® 883** può indirizzare direttamente un massimo di 1M Byte di memoria che coincide con lo spazio d'indirizzamento fisico del microprocessore. Questa capacità di memoria, sulla scheda, può essere suddivisa in diversi segmenti distinti ognuno dei quali può avere indirizzi d'inizio e dimensioni programmabili via software. La circuiteria di CSU, interna al microprocessore, si occupa appunto di dividere lo spazio direttamente indirizzato dalla CPU in vari segmenti e di allocarli in memoria nello spazio dei dispositivi fisici. Programmando la circuiteria di CSU tramite gli appositi registri UMCS, MMCS, LMCS, MPCS è quindi possibile indirizzare direttamente la quantità di memoria effettivamente montata su IC 4 ed IC 5, con la facoltà di definire anche le modalità e le tempistiche d'accesso.
Viene di seguito riportata una figura che illustra le possibili configurazioni dei dispositivi allocati nello spazio di memoria; per maggiori informazioni sulle modalità di gestione della CSU fare riferimento all'appendice B, mentre per una facile individuazione e configurazione dei dispositivi di memoria fare riferimento alle figure 24, 35.
All'atto del power on o del reset la CSU é programmata in modo da allocare un segmento di 64K Bytes alla fine dello spazio fisico d'indirizzamento della CPU, quindi la scheda parte con l'esecuzione del codice posto all'indirizzo fisico FFFF0H che corrisponde sempre con l'ultimo segmento da 16 Bytes della EPROM o FLASH EPROM di IC 5.

Alcuni pacchetti software, come il GDOS ed il GCTR, si occupano autonomamente della gestione della circuiteria di CSU per allocare tutta la memoria fisicamente presente a bordo scheda nello spazio d'indirizzamento del microprocessore, senza interessare direttamente l'utente.

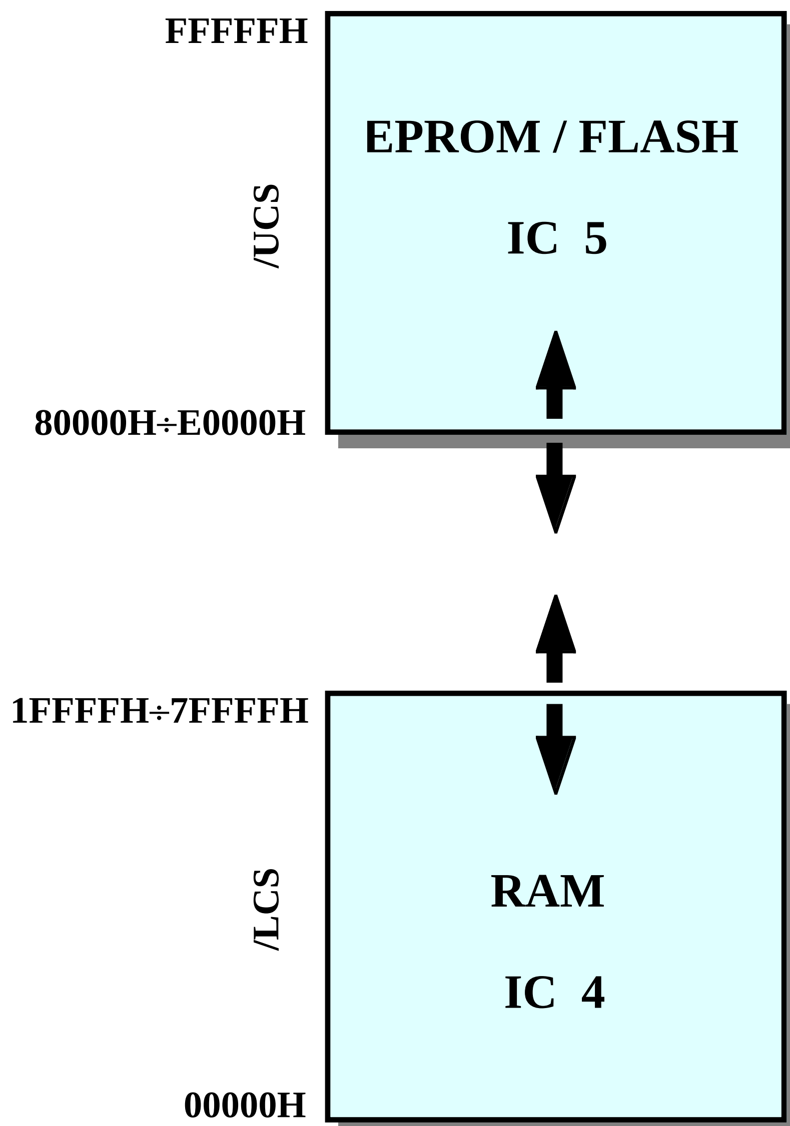


FIGURA 40: MAPPAGGIO DELLE MEMORIE

MAPPAGGIO I/O

Il mappaggio delle periferiche di bordo è avviene tramite opportuni registri allocati in altrettante locazioni di I/O. Viene di seguito riportato l'indirizzamento di tali registri il loro nome, indirizzo, il tipo di accesso ed una breve descrizione del loro significato. Per quanto riguarda la descrizione dettagliata del loro significato, si faccia riferimento al capitolo successivo “DESCRIZIONE SOFTWARE DELLE PERIFERICHE DI BORDO”.
Si ricorda che la tabella sucessiva riporta la descrizione dei soli registri delle periferiche esterne al microprocessore e che per la descrizione di quelli interni si può fare riferimento all’apposita documentazione della casa costruttrice, oppure all’appendice B di questo manuale. Il valore degli indirizzi riportati in tabella (INDPCS0, INDPCS1, ecc) é definita dalla programmazione effettuata sulla CSU ed in dettaglio degli appositi registri RELREG, MPCS, PACS.

DISP.	REG.	INDIRIZZO	R/W	SIGNIFICATO
ABACO® I/O BUS	I/OBUS	INDPCS0+00H ÷ INDPCS0+FFH	R/W	Indirizzi gestione **ABACO®** I/O BUS
JUMPER J1	RUNDEB	INDPCS1+00H	R	Registro acquisizione stato jumper J1
REAL TIME CLOCK 62421	SEC1	INDPCS1+00H	R/W	Registro dati per unità secondi
	SEC10	INDPCS1+01H	R/W	Registro dati per decine secondi
	MIN1	INDPCS1+02H	R/W	Registro dati per unità minuti
	MIN10	INDPCS1+03H	R/W	Registro dati per decine minuti
	HOU1	INDPCS1+04H	R/W	Registro dati per unità ore
	HOU10	INDPCS1+05H	R/W	Registro dati per decine ore e AM/PM
	DAY1	INDPCS1+06H	R/W	Registro dati per unità giorno
	DAY10	INDPCS1+07H	R/W	Registro dati per decine giorno
	MON1	INDPCS1+08H	R/W	Registro dati per unità mese
	MON10	INDPCS1+09H	R/W	Registro dati per decine mese
	YEA1	INDPCS1+0AH	R/W	Registro dati per unità anno
	YEA10	INDPCS1+0BH	R/W	Registro dati per decine anno
	WEE	INDPCS1+0CH	R/W	Registro dati per giorno della settimana
	REGD	INDPCS1+0DH	R/W	Registro di controllo D
	REGE	INDPCS1+0EH	R/W	Registro di controllo E
	REGF	INDPCS1+0FH	R/W	Registro di controllo F
A/D MAX197	DASCTRL	INDPCS5+00H	W	Registro di controllo A/D converter
	DASL	INDPCS5+00H	R	Registro dati low A/D converter
	DASH	INDPCS5+01H	R	Registro dati high A/D converter
PPI 82C55	PA	INDPCS5+40H	R/W	Registro dati del port A
	PB	INDPCS5+41H	R/W	Registro dati del port B
	PC	INDPCS5+42H	R/W	Registro dati del port C
	RC	INDPCS5+43H	R/W	Registro di controllo e comando
W. DOG	RWD	INDPCS5+80H	R/W	Registro di retrigger del watch dog
DIP SWITCH	DSW1	INDPCS5+C0H	R	Registro di acquisizione dip switch
CAN controller SJA 1000	CAN	INDPCS6+00H ÷ INDPCS6+7FH	R/W	Registri per la gestione dell'UART CAN SJA 1000, in modalità BasicCAN o PeliCAN (i registri sono gli stessi riportati nel data sheet del componente, con un offset INDPC6).

FIGURA 41: TABELLA INDIRIZZAMENTO I/O

DESCRIZIONE SOFTWARE DELLE PERIFERICHE DI BORDO

Nel paragrafo precedente sono stati riportati gli indirizzi di allocazione di tutte le periferiche e di seguito viene riportata una descrizione dettagliata della funzione e del significato dei relativi registri e o linee (al fine di comprendere le successive informazioni, fare sempre riferimento alle tabelle delle figure 39 e 41). Qualora la documentazione riportata fosse insufficiente fare riferimento direttamente alla documentazione tecnica della casa costruttrice del componente. In questo paragrafo inoltre non vengono descritte le sezioni che fanno parte del microprocessore; per quanto riguarda la programmazione di quest'ultime si faccia riferimento all'appendice B di questo manuale. Nei paragrafi successivi si usano le indicazioni **D0÷D7** e **.0÷7** per fare riferimento ai bits della combinazione utilizzata nelle operazioni di I/O.

INGRESSI DI CONFIGURAZIONE

La **GPC® 883** dispone di 9 ingressi di configurazione settabili dall'utente ed acquisibili via software, con le modalità di seguito riportate. Le possibili applicazioni di questi ingressi possono essere quelle destinate al settaggio di alcune condizioni di lavoro, alla selezione di parametri relativi al firmware di bordo, selezione della lingua, ecc.
Il dip switch DSW1 può essere accquisito effettuando una semplice operazione di input all'indiizzo di allocazione del registro DSW1. La corrispondenza tra i bit del registro e le linee del dip switch è la seguente:

DSW1.7 -> **dip switch 8**
DSW1.6 -> **dip switch 7**
DSW1.5 -> **dip switch 6**
DSW1.4 -> **dip switch 5**
DSW1.3 -> **dip switch 4**
DSW1.2 -> **dip switch 3**
DSW1.1 -> **dip switch 2**
DSW1.0 -> **dip switch 1**

La combinazione è in logica negata, ovvero il dip in **ON** fornisce lo stato logico **0** al corrispondente bit, mentre il dip in **OFF** fornisce lo stato logico **1**.
Lo stato del jumper di configurazione J1 può essere accquisito via software, effettuando una semplice operazione di input all'indirizzo di allocazione del registro RUNDEB ed esaminando il bit D7. L'acquisizione è in logica negata, ovvero:

J1 connesso -> **RUNDEB.7 = 0**
J1 non connesso -> **RUNDEB.7 = 1**

Tale jumper svolge la funzione di selettore delle modalità RUN (non connesso) o DEBUG (connesso), caratteristica di alcuni pacchetti software della **grifo®**. Il registro RUNDEB condivide l'indirizzo di altre periferiche di bordo ma questo non genera conflitti infatti l'acquisizione del jumper é un'operazione di input ed i dati D6÷D0 prelevati sono privi di significato.
Per una facile individuazione degli ingressi di configurazione sulla scheda, si vedano le figure 24 e 29.

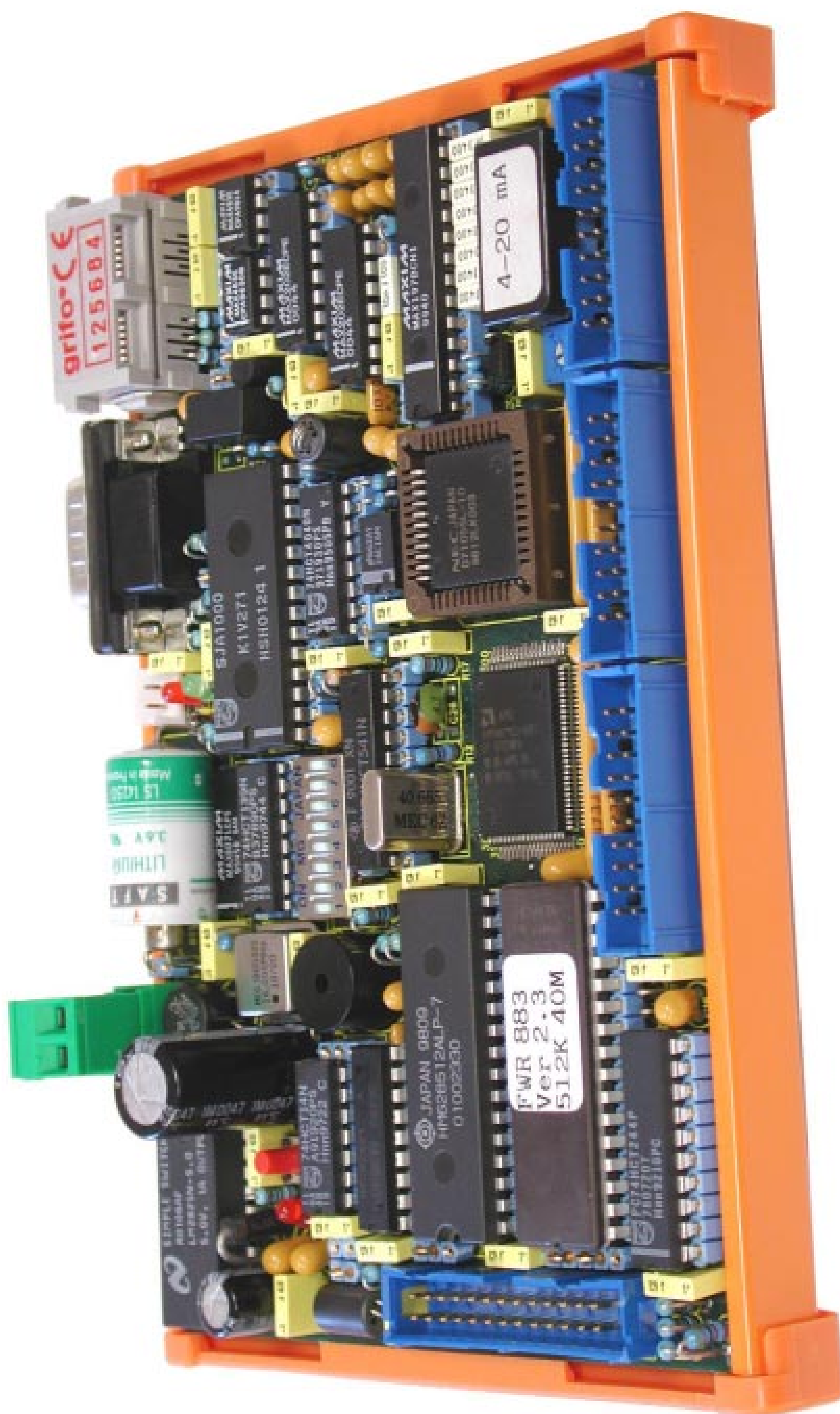


FIGURE 42: VISTA LATERALE DELLA SCHEDA CON CONTENITORE

WATCH DOG

Il retrigger della circuiteria di watch dog presente sulla **GPC® 883**, avviene tramite una semplice operazione di input e/o otput al registro RWD. Affinché la circuiteria di watch dog non intervenga, é indispensabile retriggerarla ad intervalli regolari di durata inferiore al tempo d’intervento. Se ciò non avviene e tramite il jumper J7 la circuiteria é attivata, una volta scaduto il tempo d’intervento la scheda viene resettata. Il tempo d’intervento è di circa 1420 msec (ma può variare considerevolmente da 940 a 2060 msec) e nella condizione di default, la circuiteria é disabilitata.
A seguito di un reset o power on la circuiteria di watch dog é retriggerata, in modo da lasciare il tempo al programma applicativo di essere eseguito e di effettuare quindi autonomamente il retrigger.

EEPROM SERIALE

Per quanto riguarda la gestione del modulo di EEPROM seriale (IC 22), si faccia riferimento alla documentazione specifica del componente od ai programmi dimostrativi forniti con la scheda. L'utente deve realizzare una comunicazione sincrona con il protocollo standard I^2C BUS, tramite alcune linee di I/O della sezione PIO della CPU. Le uniche informazioni necessarie sono i collegamenti elettrici:

PIO 04 (input/output)	->	linea **DATA**	**(SDA)**
PIO 29 (output)	->	linea **CLOCK**	**(SCL)**

Data l'implementazione hardware della circuiteria di gestione del modulo di EEPROM seriale, si ricorda che su tale dispositivo i segnali **A0**,**A1**,**A2** dello slave address sono tutti posti a **0** logico. Lo stato logico 0 dei bit corrisponde allo stato logico basso (=0 V) del relativo segnale, mentre lo stato logico 1 dei bit corrisponde allo stato logico alto (=5 V) del segnale.
Per ulteriori informazioni sulle modalità di gestione dei segnali della sezione PIO fare riferimento all'apposita documentazione tecnica dell'appendice B.
A seguito di un reset o power on le due linee di gestione dell'EEPROM seriale sono settate entrambe in input ed aun livello logico alto.

STATO DELLA BATTERIA

Lo stato della circuiteria di back up della **GPC® 883** può essere acquisito via software, effettuando una semplice acquisizione del segnale PIO 26 della sezione PIO, che ha la seguente corrispondenza:

PIO 26 = 0	->	batteria scarica	(**<2,265 V)**
PIO 26 = 1	->	batteria carica	(> **2,265 V**)

Come descritto nel paragrafo "BACK UP" sia la batteria di bordo che la batteria esterna sono collegate alla circuiteria di back up, quindi lo stato descritto é relativo ad entrambe. Quando le batterie sono entrambe scollegate (CN2 e J6 non collegati) lo stato della batteria può variare in modo randomico, ma questa é una condizione non significativa.
Per ulteriori informazioni sulla batteria di bordo e della relativa circuiteria di back up fare riferimento ai precedenti appositi paragragfi.

PPI 82C55

Questa periferica è vista in 4 registri: uno di controllo (RC) e tre dei dati (PA, PB, PC) con cui si effettua la programmazione ed il comando della stessa. I registri dati sono utilizzati sia per operazioni di lettura (acquisizione linee dei port) che per quelle di scrittura (settaggio linee dei port) ed ognuno di tali registri riporta i dati di I/O del corrispondente port. La periferica può operare in tre modi diversi:

MODO 0 = Prevede due port bidirezionali da 8 bit (A,B) e due port bidirezionali da 4 bit (C LOW, C HIGH); gli ingressi non sono latchati, mentre le uscite lo sono; nessun segnale di handshaking.
MODO 1 = Prevede due port da 12 bit (A+C LOW, B+C HIGH) dove gli 8 bit dei port A e B costituiscono le linee di I/O, mentre i 4 bit del port C costituiscono le linee di handshaking. Gli ingressi e le uscite sono latchati.
MODO 2 = Prevede un port da 13 bit (A+C3-7) dove gli 8 bit del port A costituiscono le linee di I/O, mentre i rimanenti 5 bit del port C costituiscono le linee di controllo. Un port da 11 bit (B+C0-2) dove gli 8 bit del port B costituiscono le linee di I/O ed i rimanenti 3 bit del port C costituiscono le linee di controllo. Sia gli ingressi che le uscite sono latchate.

La programmazione della periferica avviene scrivendo un byte nel registro di controllo RC, settando gli 8 bit del dato scritto con la seguente corrispondenza:

		D7	D6	D5	D4	D3	D2	D1	D0
RC	**=**	**SF**	**M1**	**M2**	**A**	**CH**	**M3**	**B**	**CL**

dove:

SF	= Se attivo (1) abilita il comando della periferica
M1 M2	= Selezionano il modo di funzionamento
0 0	= Selezione del modo 0
0 1	= Selezione del modo 1
1 X	= Selezione del modo 2
A	= Se attivo (1) setta il port A in input e viceversa
CH	= Se attivo setta il nibble più significativo del port C in input e viceversa
M3	= Se attivo (1) seleziona modo 1, viceversa seleziona modo 0
B	= Se attivo setta il port B in input e viceversa
CL	= Se attivo setta il nibble meno significativo del port C in input e viceversa.

Dopo una fase di reset o di power on il PPI 82C55 viene settato in modo 0 con tutti i port in input.

LED DI ATTIVITA'

La logica di controllo consente la gestione software di un LED di attività, chiamato LD3, tramite il segnale PIO 06 della sezione PIO, con le seguenti corrispondenze:

PIO 06 = 0 -> **LD3 disattivo**
PIO 06 = 1 -> **LD3 attivo**

Il segnale PIO 06 è settato a livello logico 0 in fase di reset o power on, di conseguenza in seguito ad una di queste fasi il LED è disattivo.

REAL TIME CLOCK

Questa periferica è vista in 16 locazioni di I/O consecutive di cui 3 di stato e le rimanenti 13 per i dati. I registri dati sono utilizzati sia per operazioni di lettura (dell'orario attuale) che di scrittura (per il settaggio dell'orologio) così come i registri di stato e controllo i quali sono utilizzati in scrittura (per la programmazione del modo di funzionamento dell'orologio) ed in lettura (per determinare lo stato dell'orologio).
Per quanto riguarda il significato dei registri dati vale la corrispondenza:

SEC1	- Unita' dei secondi	- 4 bit meno significativi:	**SEC1.3÷SEC1.0**
SEC10	- Decine dei secondi	- 3 bit meno significativi:	**SEC10.2÷SEC10.0**
MIN1	- Unita' dei minuti	- 4 bit meno significativi:	**MIN1.3÷MIN1.0**
MIN10	- Decine dei minuti	- 3 bit meno significativi:	**MIN10.2÷MIN10.0**
HOU1	- Unita' delle ore	- 4 bit meno significativi:	**HOU1.3÷HOU1.0**
HOU10	- Decine delle ore	- 2 bit meno significativi:	**HOU10.1÷HOU10.0**
		Il terzo bit di tale registro, **HOU10.2**, indica l'AM/PM	
DAY1	- Unita' del giorno	- 4 bit meno significativi:	**DAY1.3÷DAY1.0**
DAY10	- Decine del giorno	- 2 bit meno significativi:	**DAY10.1÷DAY10.0**
MON1	- Unita' del mese	- 4 bit meno significativi:	**MON1.3÷MON1.0**
MON10	- Decine del mese	- 1 bit meno significativo:	**MON10.0**
YEA1	- Unita' dell'anno	- 4 bit meno significativi:	**YEA1.3÷YEA1.0**
YEA10	- Decine dell'anno	- 4 bit meno significativi:	**YEA10.3÷YEA10.0**
WEE	- Giorno della settimana	- 3 bit meno significativi:	**WEE.2÷WEE.0**

Per quest'ultimo registro vale la corrispondenza:

WEE.2	**WEE.1**	**WEE.0**	Giorno della settimana
0	0	0	Domenica
0	0	1	Lunedi'
0	1	0	Martedi'
0	1	1	Mercoledi'
1	0	0	Giovedi'
1	0	1	Venerdi'
1	1	0	Sabato

I tre registri di controllo sono invece utilizzati come segue:

	D7	D6	D5	D4	D3	D2	D1	D0
REGD =	**NU**	**NU**	**NU**	**NU**	**30S**	**IF**	**B**	**H**

dove:

NU = Non usato

30S = Se attivo (1) permette di effettuare una correzione di 30 secondi dell'orario. Una volta settato i secondi del RTC vengono azzerati ed i minuti incrementati se il precedente valore dei secondi era superiore o uguale a 30.

IF = Gestisce lo stato d’interrupt del RTC. In lettura riporta lo stato attuale d'interrupt (1=attivo e viceversa), mentre se resettato con una scrittura determina la fine interrupt, quando il RTC lavora in interrupt mode.

B = Indica se possono essere effettuate operazioni di lettura/scrittura dei registri:
1 -> operazioni impossibili e viceversa.

H = Se attivo (1) effettua la memorizzazione dell'orario fissato.

	D7	D6	D5	D4	D3	D2	D1	D0
REGE =	**NU**	**NU**	**NU**	**NU**	**T1**	**T0**	**I**	**M**

dove:

NU = Non usato.

T1 T0 = Determinano la durata del periodo di interrupt

T1	T0		
0	0	->	1/64 secondo
0	1	->	1 secondo
1	0	->	1 minuto
1	1	->	1 ora

I = Determina modalità di gestione interrupt:
- 1 -> seleziona l'interrupt mode in cui l'interrupt si attiva allo scadere del periodo programmato e si disattiva con un reset del bit IF del registro D;
- 0 -> seleziona lo standard mode in cui l'interrupt si attiva allo scadere del periodo programmato e si disattiva autonomamente dopo 7,8 msec.

M = Maschera la generazione d'interrupt:
- 1 -> interrupt mascherato, il pin di interrupt é sempre disattivo.
- 0 -> interrupt non mascherato, il pin di interrupt riflette lo stato d'interrupt.

	D7	D6	D5	D4	D3	D2	D1	D0
REGF =	**NU**	**NU**	**NU**	**NU**	**T**	**24/12**	**S**	**R**

dove:

NU = Non usato.

T = Stabilisce da quale contatore interno prelevare il segnale di conteggio:
- 1 -> contatore principale (conteggio veloce per test);
- 0 -> 15° contatore (conteggio normale).

24/12 = Stabilisce il modo di conteggio delle ore:
- 1 -> 0÷23;
- 0 -> 0÷11 con AM/PM.

S = Se settato (1) provoca l'arresto dell'avanzamento dell'orologio fino alla sucessiva abilitazione (0).

R = Se settato (1) provoca il reset di tutti i contatori interni.

I registri del RTC utilizzano lo stesso indirizzo del registro RUNDEB, quindi ogni accesso ai registri del RTC implica anche l'acquisizione dell'ingresso di configurazione J1.
A seguito di un reset o power on nessuno dei registri del RTC é settato, mantenendo quindi i precedenti valori; se la circuiteria di back up é collegata i registri dati del RTC sono aggiornati anche quando la scheda non é alimentata.

BUZZER

Il buzzer BZ1 é gestito via software tramite il segnale PIO 25 della sezione PIO, con le seguenti corrispondenze:

PIO 25 = 0 -> **BZ1 attivo**
PIO 25 = 1 -> **BZ1 disattivo**

La funzione principale del buzzer é quella di segnalare acusticamente delle condizioni di stato previste dal programma applicativo in modo da attirare l'attenzione dell'utilizzatore (ad esempio allarmi, configurazioni anomale, ecc).
Il segnale PIO 25 è settato a livello logico 1 in fase di reset o power on, di conseguenza in seguito ad una di queste fasi il buzzer è disattivo.

PERIFERICHE DELLA CPU

La descrizione dei registri e del relativo significato di tutte le periferiche interne della CPU (ASP, TCU, DMA, ICU, CSU, ecc) é disponibile nell'appendice B. Qualora queste informazioni fossero ancora insufficienti, fare riferimento alla documentazione tecnica della casa costruttrice.

CONTROLLORE CAN

Come indicato nella tabella di figura 41 il controllore CAN **SJA 1000**, é gestito tramite una serie di registri di comando e/o stato che sono approfonditamente descritti nell'appendice B di questo manuale ed utilizzati negli esempi forniti assieme alla scheda. In questo paragrafo vengono solo riportate una serie di informazioni aggiuntive necessarie per usare correttamente tali registri.

1) Il Bit Rate di comunicazione, ricavabile dalle informazioni dell'appendice B, é ottenibile dalla seguente formula:

BAUD RATE = Freq / 2 * (BRP + 1) * (3 + TSEG1 + TSEG2)

dove:
Freq = Frequenza di clock del controllore CAN in Hz (16000000).
BRP = Valore espresso dai bit **BRP.x** del **Bus Timing Register 0** (BTR0, indirizzo **INDPCS6+6**).
TSEG1 = Valore espresso dai bit **TSEG1.x** del **Bus Timing Register 1** (BTR1, indirizzo **INDPCS6+7**).
TSEG2 = Valore espresso dai bit **TSEG2.x** del **Bus Timing Register 1** (BTR1, indirizzo **INDPCS6+7**).

2) Per un corretto interfacciamento fra il controllore CAN **SJA1000** ed il driver di linea **82C250**, é necessario programmare l'**Output control register** (OCR, indirizzo **INDPCS6+8**), con il dato **FA Hex**; in questo modo si configura il dispositivo in "Normal output mode", con le uscite TX0 e TX1 in "Push-Pull".

3) Nelle figura seguente viene riportata la flow chart relativa all'inizializzazione consigliata del controllore CAN **SJA1000**: come si può notare, questa inizializzazione non prevede l'utilizzo di Interrupt. Per una loro eventuale gestione, é necessario settare opportunamente i bit del Control Register (CR, indirizzo **INDPCS6**+0).

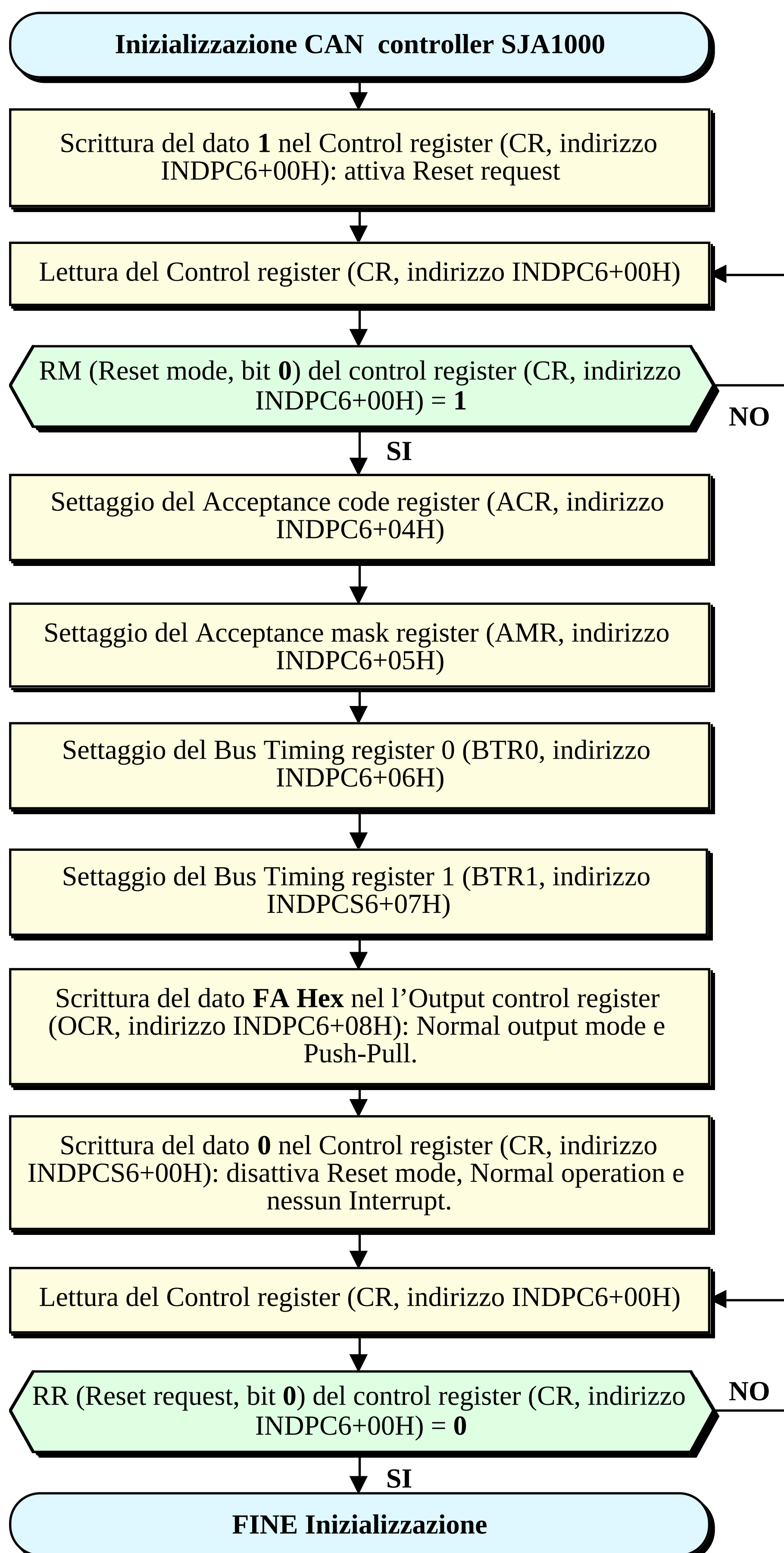


FIGURA 43: FLOW CHART INIZIALIZZAZIONE CONTROLLORE CAN

A/D CONVERTER MAX 197

La sezione di A/D converter della **GPC® 883** é basata sul DAS MAX 197; tale dispositivo é gestito tramite tre registri, uno di scrittura e due di lettura denominati DASCTRL, DASL e DASH. Di seguito é riportato il significato di questi registri ed il loro utilizzo.

	D7	D6	D5	D4	D3	D2	D1	D0
DASCTRL =	**PD1**	**PD0**	**ACQMOD**	**RNG**	**BIP**	**A2**	**A1**	**A0**

Questo registro di scrittura gestisce il controllo del DAS MAX 197: una operazione di scrittura in questo indirizzo farà partire la conversione, nel canale e nel range specificato.
Il significato dei bit che compongono tali registri é il seguente:

PD1 PD0 = Settano il modo di funzionamento dell'A/D converter tra le seguenti modalità:

PD1	PD0	
0	0	-> Funzionamento normale con clock esterno
0	1	-> Funzionamento normale con clock interno (NON USARE)
1	0	-> Modalità Standby Power Down
1	1	-> Modalità Full Power Down

ACQMOD = Controlla il funzionamento del Track Hold interno all'A/D:

0 -> Il Track Hold del canale specificato da A0÷A2, rimane attivo per 3 µsec quindi inizia la fase di conversione del segnale catturato.

1 -> Il Track Hold del canale specificato da A0÷A2, si attiva e rimane in questo stato fino alla prossima scrittura nel registro di controllo, il cui dato dovrà essere uguale al precedente, ma con ACQMOD=0. Utilizzando questà modalità l'utente può catturare il segnale da convertire per il tempo da lui desiderato.

RNG BIP = Settano il range di tensione del canale analogico, specificato dai bit A0÷A2; tale configurazione é relativa solo alla conversione che l'A/D converter andrà ad effettuare, quindi può variare da canale a canale ed anche su uno stesso, in corrispondenza di ogni nuova conversione:

RNG	BIP	
0	0	-> Range 0÷5 V
1	0	-> Range 0÷10 V
0	1	-> Range ±5 V
1	1	-> Range ±10 V

A2 A1 A0 = Selezionano il canale analogico d'ingresso sul quale sarà eseguita l'operazione di Track Hold e la successiva conversione del segnale catturato:

A2	A1	A0	
0	0	0	-> Canale 0
0	0	1	-> Canale 1
0	1	0	-> Canale 2
0	1	1	-> Canale 3
1	0	0	-> Canale 4
1	0	1	-> Canale 5
1	1	0	-> Canale 6
1	1	1	-> Canale 7

N.B.
La modalità di funzionamento normale con clock Interno, **non deve essere mai utilizzata**, in quanto la scheda **GPC® 883** possiede un'apposita circuiteria per la generazione della frequenza di clock in modo da ottimizare, al massimo, sia il tempo di conversione che l'immunità al rumore del campo.

A titolo di esempio, scrivendo il dato 0BH nel registro ADCNT, si eseguirà una conversione della linea analogica CH3 collegata al connettore CN6, nel range ±5 V con modalità di Track Hold automatica.

D7 D6 D5 D4 D3 D2 D1 D0
DASL = C7 C6 C5 C4 C3 C2 C1 C0
Questo registro di lettura permette di acquisire il byte basso della combinazione determinata dall'A/D converter; una operazione di lettura su questo indirizzo restituirà i bit 7÷0 della combinazione relativa all’ultima conversione eseguita sul canale selezionato.
Da notare che questo registro contiene una combinazione valida solo quando la sezione A/D non stà eseguendo la fase di Track Hold o quella di conversione, ovvero a conversione ultimata.

D7 D6 D5 D4 D3 D2 D1 D0
DASH = NU NU NU NU C11 C10 C9 C8
Questo registro di lettura permette di acquisire il byte alto della combinazione determinata dall'A/D converter; una operazione di lettura su questo indirizzo restituirà i bit 11÷8 della combinazione relativa all'ultima conversione eseguita sul canale selezionato.
Da notare che questo registro contiene una combinazione valida solo quando la sezione A/D non stà eseguendo la fase di Track Hold o quella di conversione, ovvero a conversione ultimata.
Il bit D3=C11 conterrà il docicesimo bit della combinazione nel caso delle conversioni nei range unipolari (0÷5 V, 0÷10 V) oppure il segno della stessa (**0** = positivo, **1** = negativo) nel caso delle conversioni nei range bipolari (±5 V, ±10 V). Questo tipo di codifica si presta facilmente ad essere utilizzata nella maggioranza dei linguaggi di programmazione, infatti la tecnica che questi usano per codificare i dati interi con segno é compatibile con la modalità descritta. Per maggiori informazioni si vedano gli appositi programmi di esempio forniti in abbinamento alla scheda.
Da ricordare che l'accesso ad ognuno dei tre registri della scheda, implica anche la disattivazione del segnale di interrupt della **ADC 812**.

A titolo di esempio se a seguito di una conversione con range ±5 V dal registro DASL viene acquisito il dato 40H e dal registro DASH il dato 90H la combinazione complessiva risultante é: -140H = -320.
A scopo delucidativo, viene di seguito riportata una breve descrizione di come gestire una conversione sfruttando la sezione A/D converter della **GPC® 883**:

SEQUENZA DI INIZIALIZZAZIONE

Dopo un’operazione di power on della **GPC® 883**, la sezione A/D converter di bordo si trova in uno stato di azzeramento generale ed é pronta a ricevere una scrittura nel registro **DASCTRL** per effettuare una conversione. Nel caso la scheda di controllo sia semplicemente resettata (ad esempio per l'utilizzo della circuiteria di watch dog od un reset manuale) la **GPC® 883** non viene resettata e mantiene il suo precedente stato. Questa situazione non é comunque problematica in quanto la prima scrittura sul registro **DASCTRL** azzererà la condizione preesistente attivando contemporaneamente una nuova conversione.
La scheda non necessita quindi di particolari procedure di inizializzazione.

CONVERSIONE IN POLLING

Di seguito viene illustrata la sequenza di operazioni da effettuare per eseguire una conversione su un canale della **GPC® 883**. La fine conversione viene controllata in modo ciclico (Polling) per attendere la possibilità di prelevare la combinazione.

- Scrittura nel registro **DASCTRL** del dato relativo alle specifiche desiderate (canale, range, Track Hold, ecc..)
- Attesa di un tempo pari a quello di conversione ($\geq$ 9 µsec)
- Lettura dei bit 11÷8 della combinazione relativa alla conversione eseguita dal registro **DASH**
- Lettura dei bit 7÷0 della combinazione relativa alla conversione eseguita dal registro **DASL**
- Elaborazione della combinazione ottenuta

CONVERSIONE IN INTERRUPT

Di seguito viene illustrata la sequenza di operazioni da effettuare per eseguire una conversione su un canale della **GPC® 883**. Il segnale di fine conversione causerà un'evento d'interrupt che indicherà alla CPU che la combinazione é disponibile e può essere prelevata.

Il programma principale deve effettuare le seguenti operazioni:
- Scrittura nel registro **DASCTRL** del dato relativo alle specifiche desiderate (canale, range, Track Hold, ecc..)
- Elaborazione della combinazione restituita dalla procedura di risposta all’interrupt

Mentre la procedura di riposta all’interrupt deve contenere:
- Lettura dei bit 11÷8 della combinazione relativa alla conversione eseguita dal registro **DASH**
- Lettura dei bit 7÷0 della combinazione relativa alla conversione eseguita dal registro **DASL**

SCHEDE ESTERNE

La scheda **GPC® 883** si interfaccia a buona parte dei moduli della serie BLOCK e di interfaccia operatore della **grifo®** od a molti altri sistemi di altre ditte. Le risorse di bordo possono essere facilmente aumentate collegandola alle numerose schede periferiche del carteggio **grifo®** tramite i suoi connettori standard. Anche schede in formato Europa con BUS **ABACO®** possono essere collegate, sfruttando gli appositi mother boards. A titolo di esempio ne riportiamo un elenco con una breve descrizione delle carratteristiche di massima; per maggiori informazioni richiedere o cercare l'apposita documentazione tecnica, sul CD o sul sito **grifo®**.

QTP G28

Quick Terminal Panel 28 tasti con LCD grafico

Interfaccia operatore provvista di display LCD grafico da 240x128 pixel retroilluminato con lampada a catodo freddo; tastiera a membrana da 28 tasti di cui 6 configurabili dall'utente; 16 LEDs di stato con attributo di lampeggio; alimentatore a bordo scheda; interdaccia seriale in RS 232, RS 422-485 o current loop; linea seriale ausiliaria in RS 232; controllore per linea CAN. Tasti ed etichette personalizzabili dall'utente tramite serigrafie da inserire in apposite tasche; contenitore metallico e plastico; EEPROM di set up; 256K EPROM o FLASH; Real Time Clock; 128K RAM; buzzer. Firmware di gestione che svolge funzione di terminale con primitive grafiche.

ADC 812

Analog to Digital Converter, 8 channels, 12 bits multi-range

Modulo periferico della serie 4 (100x50 mm); A/D converter DAS (Data Acquisition System) multi-range a 8 canali da 12 bit; Track-Hold; tempo di conversione 9µs; range dei segnali d'ingresso ±10, ±5, +10, +5Vdc oppure 0÷20, 4÷20mA; interfaccia per **ABACO®** I/O BUS; possibilità di montaggio diretto su guide Ω di tipo DIN 46277-1 e 3.

DAC 212

Digital to Analog Converter 12 bits, multi-range

Modulo periferico della serie 4 (100x50 mm); D/A converter multi-range a 2 canali da 12 bit; range del segnali d'uscita ± 10 o 0/+10 Vdc; interfaccia per **ABACO®** I/O BUS; possibilità di montaggio diretto su guide Ω di tipo DIN 46277-1 e 3.

CAN 14

Control Area Network, 1 channel, galvanically insulated

Modulo periferico della serie 4 (100x50 mm); UART CAN SJA1000; 1 canale seriale galvanicamente isolato; interfaccia per **ABACO®** I/O BUS; possibilità di montaggio diretto su guide Ω di tipo DIN 46277-1 e 3.

IBC 01

Interface Block Comunication

Scheda di conversioni per comunicazioni seriali. 2 linee RS 232; 1 linea RS 422-485; 1 linea in fibra ottica; interfaccia DTE/DCE selezionabile; attacco rapido per guide tipo DIN 46277-1 e 3.

KDL X24 - KDF 224

Keyboard Display LCD 2,4 righe 24 tasti - Keyboard Display Fluorescent 2 righe 24 tasti

Interfaccia tra 16 I/O TTL su connettore normalizzato I/O **ABACO®** a 20 vie e tastiera a matrice esterna da 24 tasti; display alfanumerico fluorescente 20x 2 o LCD 20x2, 20x4 retroilluminato a LEDs. Predisposizione per collegamento a tastiera telefonica.

QTP 24P

Quick Terminal Panel 24 tasti con interfaccia Parallela

Interfaccia operatore provvista di display alfanumerico fluorescente 20x 2 o LCD 20x2, 20x4 retroilluminato a LEDs; tastiera a membrana da 24 tasti di cui 12 configurabili dall'utente; 16 LEDs di stato; alimentatore a bordo scheda in grado di pilotare anche carichi esterni; interdaccia parallela basata su 16 I/O TTL di un connettore normalizzato I/O **ABACO®** a 20 vie. Tasti ed etichette personalizzabili tramite serigrafie da inserire in apposite tasche; opzione di contenitore metallico.

MCI 64

Memory Cards Interfaces 64 MBytes

Interfaccia per la gestione di Memory cards PCMCIA a 68 pins tramite un connettore normalizzato I/O **ABACO®**; sono disponibili driver per linguaggi ad alto livello.

ZBR xxx

Zipped BLOCK Relays xx Input + xx Output

Periferica per xx Input optoisolati e visualizzati tipo NPN; xx relé da 3A con MOV; connettori a morsettiera per ingressi optoisolati e uscite; connettore normalizzato **ABACO®** I/O BUS; LEDs di visualizzazione; sezione alimentatrice a bordo; attacco rapide per guide Ω. Le possibili configurazioni in termini di numero di I/O sono: xxx=324 con 32 In e 24 Out; xxx=246 con 24 In e 16 Out; xxx=168 con 16 In e 8 Out; xxx=84 con 8 In e 4 Out.

FBC xxx

Flat Block Contact xxx vie

Interfaccia tra 1 connettore a perforazione di isolante (scatolino da xx vie maschio) e la filatura da campo (morsettiere a rapida estrazione). Attacco rapido per guide tipo DIN 46277-1 e 3.

OBI N8 - OBI P8

Opto BLOCK Input NPN-PNP

Interfaccia per 8 input optoisolati e visualizzati tipo NPN, PNP, connettore a morsettiera, connettore standard I/O **ABACO®** a 20 vie; sezione alimentatrice; attacco rapido per guide DIN 46277-1 e 3.

TBO 01 - TBO 08

Transistor BLOCK Output

Interfaccia per 16 connettore normalizzato I/O **ABACO®** a 20 vie; 16 o 8 output a transistor in Open Collector da 45 Vcc 3 A su connettore a morsettiera. Uscite optoisolate e visualizzate; attacco rapido per guide DIN 6277-1 e 3.

ABB 03

Abaco® Block BUS 3 slots

Mother board **ABACO®** da 3 slots; passo 4 TE; guidaschede; connettori normalizzati di alimentazione; tasto di reset; LEDs per alimentazioni; interfaccia **ABACO®** I/O BUS. Attacco rapido per guide Ω.

IPC 52

Intelligent Peripheral Controller

Scheda periferica intelligente in grado di acquisire 24 segnali analogici generati da trasduttori da campo; 8 ingressi per PT 100, PT 1000; 8 ingressi per termocoppie J,K,S,T; 8 ingressi per segnali in tensione ±2 V o corrente 0÷20 mA; interrogazione tramite BUS **ABACO®** o tramite linea seriale in RS 232, RS 422-485 o current loop; 16 linee di I/O TTL; risoluzione di 16 bit più segno; 0,1 °C di precisione; 5 acquisizioni al secondo; funzionamento come data logher.

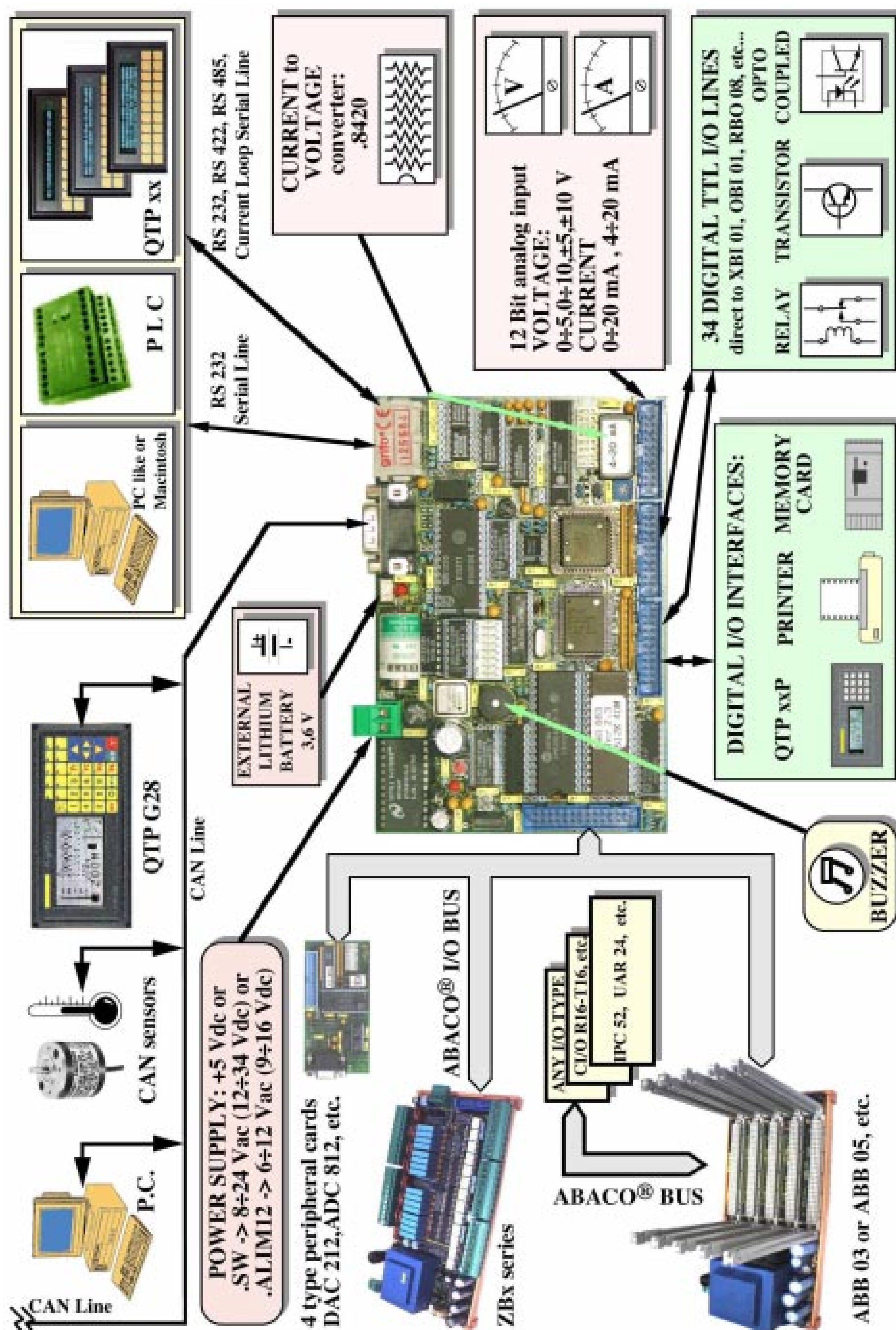


FIGURA 44: SCHEMA DELLE POSSIBILI CONNESSIONI

BIBLIOGRAFIA

E' riportato di seguito, un elenco di manuali e note tecniche, a cui l'utente può fare riferimento per avere maggiori chiarimenti, sui vari componenti montati a bordo della scheda **GPC® 883**.

Manuale AMD	*Flash Memory Products*
Manuale AMD:	*Am186ES and Am188ES User manual*
Manuale HEWLETT PACKARD:	*Optoelectronics Designer's Catalog*
Manuale MAXIM:	*New Releases Data Book - Volume IV*
Manuale MAXIM:	*New Releases Data Book 1996 - Volume V*
Manuale NATIONAL SEMICONDUCTOR:	*Databook - Linear 1*
Manuale NATIONAL SEMICONDUCTOR:	*Databook - Linear 2*
Manuale NEC:	*Memory Products*
Manuale NEC:	*Microprocessors and Peripherals - Volume 3*
Manuale NEWPORT:	*DC-DC Converters*
Manuale PHILIPS:	*Application notes and development tools for 80C51 microcontrollers*
Documentazione SEIKO EPSON:	*RTC 62421 Real Time Clock module*
Manuale TEXAS INSTRUMENTS:	*The TTL Data Book - SN54/74 Families*
Manuale TEXAS INSTRUMENTS:	*RS-422 and RS-485 Interface Circuits*
Manuale TOSHIBA:	*Photo couplers Data Book*
Manuale XICOR:	*Data Book*

Per reperire questi manuali fare riferimento alle case produttrici ed ai relativi distributori locali. In alternativa si possono ricercare le medesime informazioni o gli eventuali aggiornamenti ai siti internet delle case elencate.

APPENDICE A: MONTAGGIO MECCANICO DELLA SCHEDA

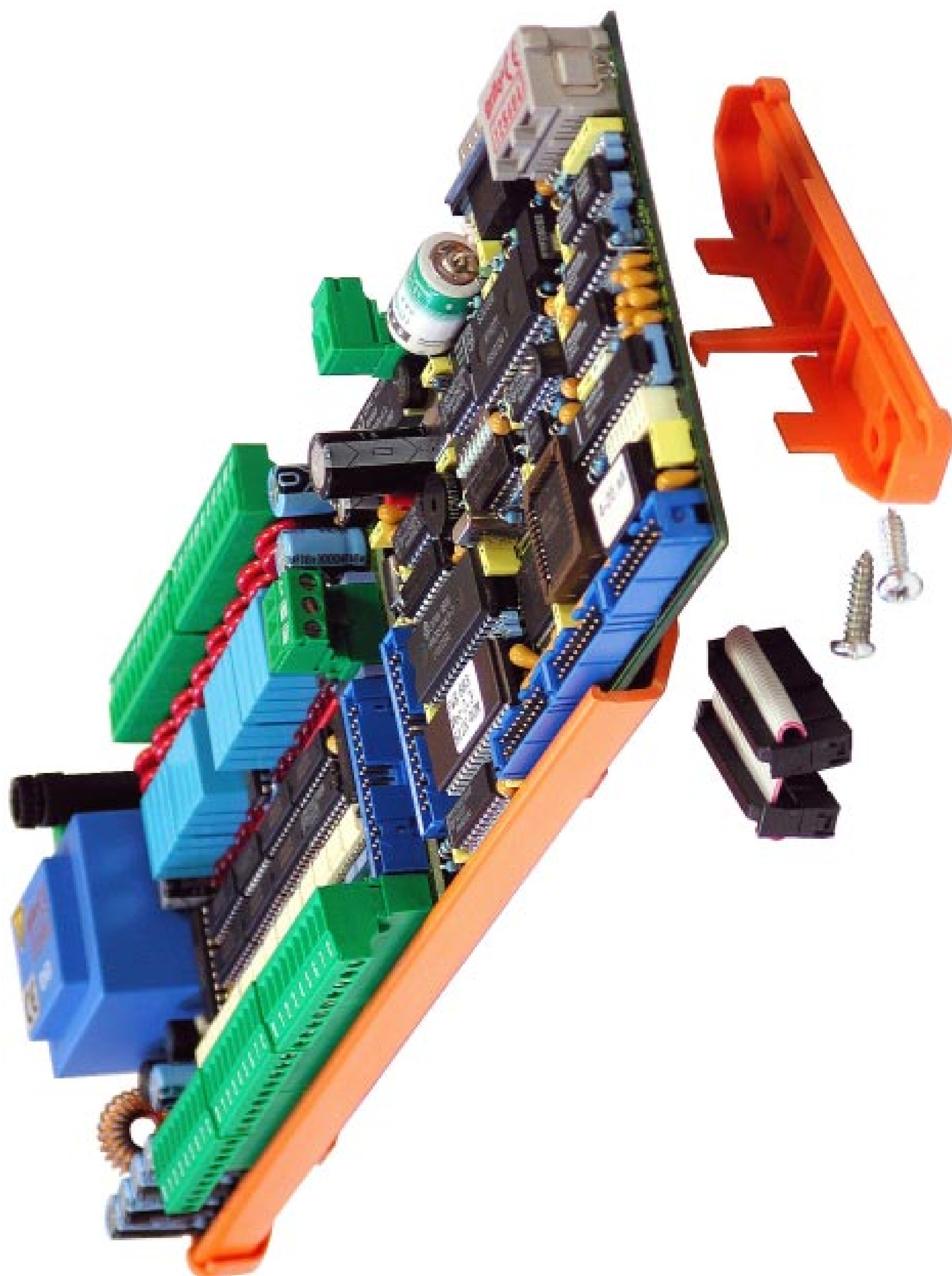


FIGURA A1: MONTAGGIO SU GUIDA WEIDMULLER

La notevole completezza della **GPC® 883** ha comportato una notevole copertura di tutto il suo circuito stampato e non sono quindi presenti fori per il montaggio meccanico. Comunque la scheda può essere usata in alcune diverse modalità:

- Utilizzando guida schede e relativi sistemi di bloccaggio scheda: tali elementi vanno fissati sul contenitore della scheda (quadro elettrico, scatola impermeabile, rack metallico, ecc.) e la scheda viene semplicemente infilata e trattenuta dalle stessa coppia di guide.

- Utilizzando un supporto plastico con attacchi diretti per le guide Ω DIN 46277-1 e 3: in questo caso é sufficiente disporre di una guida Ω all'interno del contenitore esterno (quadro elettrico, scatola impermeabile, ecc.) e su questa applicarvi quindi la scheda, tramite il suo supporto. Tale opzione é ordinabile con il codice **BLOCK.100.50**.
  Se la scheda é abbinata ad altre schede periferiche di larghezza 100 mm (ad esempio **ZBR xxx**, **ZBT xxx**, **DAC 212**, ecc.) può essere utilizzato un singolo supporto più lungo, ottenendo un unico elemento. Questo supporto plastico é il tipo RS/100 della Weidmuller (codice 414487), e può essere ordinato alla **grifo®** come opzione **EXT-WMlll**, dove lll indica la lunghezza desiderata, in mm.
  In questo caso il collegamento elettrico tra la **GPC® 883** e la scheda periferica avviene tramite un flat cable per i segnali dell'**ABACO®** I/O BUS, che deve essere più corto possibile, come il **FLT.26+26 I/O** della **grifo®**. Nella pagina precedente é riportata una foto che illustra e chiarisce quanto fin'ora esposto.

CPU AM188ES

PRELIMINARY



Am186™ES/ESLV and Am188™ ES/ESLV

High Performance, 80C186-/80C188-Compatible and 80L186-/80L188-Compatible, 16-Bit Embedded Microcontrollers

DISTINCTIVE CHARACTERISTICS

- **E86™ family 80C186-/188- and 80L186-/188-compatible microcontrollers with enhanced bus interface**
  - Lower system cost with higher performance
  - 3.3-V ±0.3-V operation (Am186ESLV and Am188ESLV microcontrollers)
- **High performance**
  - 20-, 25-, 33-, and 40-MHz operating frequencies
  - Supports zero-wait-state operation at 25 MHz with 100-ns static memory (Am186ESLV and Am188ESLV microcontrollers) and 40 MHz with 70-ns static memory (Am186ES and Am188ES microcontrollers)
  - 1-Mbyte memory address space
  - 64-Kbyte I/O space
- **Enhanced features provide improved memory access and remove the requirement for a 2x clock input**
  - Nonmultiplexed address bus
  - Processor operates at the clock input frequency
  - On the Am186ES/ESLV microcontroller, 8-bit or 16-bit memory and I/O static bus option
- **Enhanced integrated peripherals provide increased functionality, while reducing system cost**
  - Thirty-two programmable I/O (PIO) pins
  - Two full-featured asynchronous serial ports allow full-duplex, 7-bit, 8-bit, or 9-bit data transfers
  - Serial port hardware handshaking with $\overline{\text{CTS}}$, $\overline{\text{RTS}}$, $\overline{\text{ENRX}}$, and $\overline{\text{RTR}}$ selectable for each port
  - Multidrop 9-bit serial port protocol
  - Independent serial port baud rate generators
  - DMA to and from the serial ports
  - Watchdog timer can generate NMI or reset
  - A pulse-width demodulation option
  - A data strobe, true asynchronous bus interface option included for $\overline{\text{DEN}}$
  - Pseudo static RAM (PSRAM) controller includes auto refresh capability
  - Reset configuration register
- **Familiar 80C186/80L186 peripherals**
  - Two independent DMA channels
  - Programmable interrupt controller with up to eight external and eight internal interrupts
  - Three programmable 16-bit timers
  - Programmable memory and peripheral chip-select logic
  - Programmable wait state generator
  - Power-save clock divider
- **Software-compatible with the 80C186/80L186 and 80C188/80L188 microcontrollers with widely available native development tools, applications, and system software**
- **A compatible evolution of the Am186™EM and Am188™EM microcontrollers**
- **Available in the following packages:**
  - 100-pin, thin quad flat pack (TQFP)
  - 100-pin, plastic quad flat pack (PQFP)

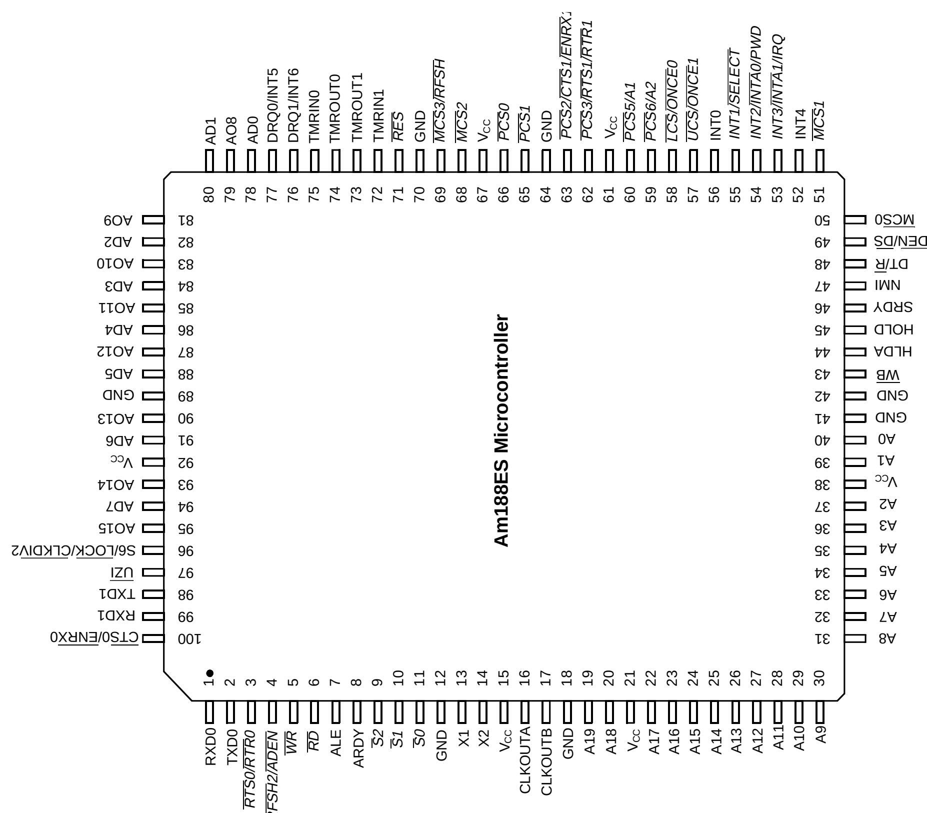
GENERAL DESCRIPTION

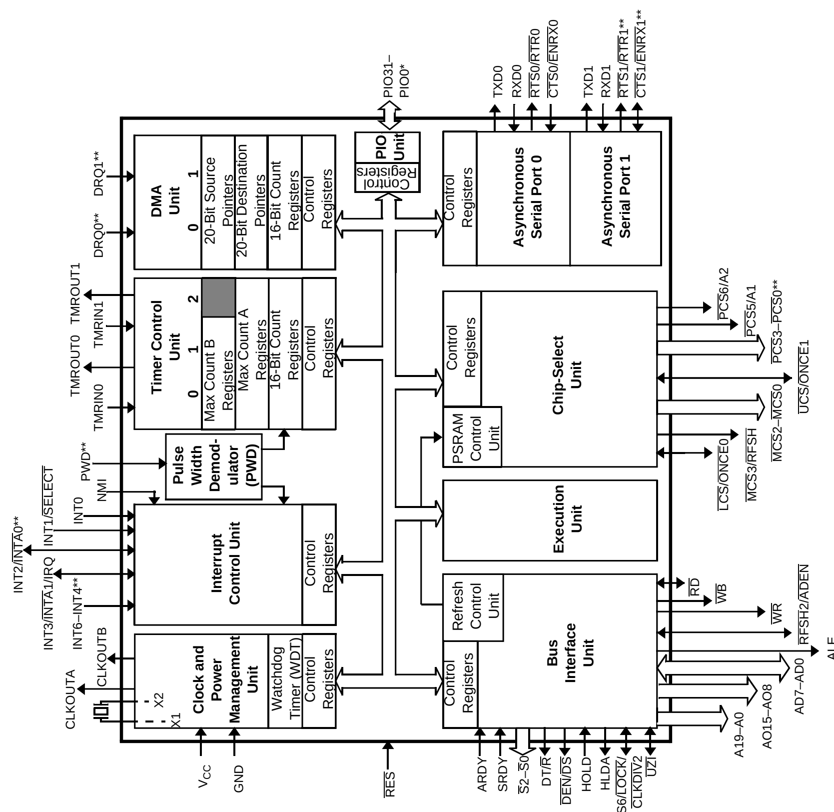
The Am186™ES/ESLV and Am188™ES/ESLV microcontrollers are an ideal upgrade for 80C186/188 and 80L186/188 microcontroller designs requiring 80C186/188 and 80L186/188 compatibility, increased performance, serial communications, and a direct bus interface.

The Am186ES/ESLV and Am188ES/ESLV microcontrollers are part of the AMD E86 family of embedded microcontrollers and microprocessors based on the x86 architecture. The E86 family includes the 16- and 32-bit microcontrollers and microprocessors described on page 8.

The Am186ES/ESLV and Am188ES/ESLV microcontrollers have been designed to meet the most common requirements of embedded products developed for the office automation, mass storage, and communications markets. Specific applications include disk drives, hand-held and desktop terminals, set-top controllers, fax machines, printers, photocopiers, feature phones, cellular phones, PBXs, multiplexers, modems, and industrial controls.

Publication# **20002** Rev: **B** Amendment/**0**
Issue Date: **February 1997**





Notes:
*All PIO signals are shared with other physical pins. See the pin descriptions beginning on page 27 and Table 2 on page 34 for information on shared functions.
** PWD, INT5, INT6, RTS1/RTR1, and CTS1/ENRX1 are multiplexed with INT2/INTA0, DRQ0, DRQ1, PCS3, and PCS2 respectively. See the pin descriptions beginning on page 27.

Am186/188ES and Am186/188ESLV Microcontrollers 3

Table 4-1 Peripheral Control Block Register Map

Register Name	Offset	Page
Processor Control Registers: Chapters 4 and 6		
Peripheral control block relocation register	FEh	4-3
Reset configuration register	F6h	4-4
Processor release level register	F4h	4-5
Auxiliary configuration register	F2h	4-6
System configuration register	F0h	4-7
Watchdog timer control register	E6h	6-3
Enable RCU register	E4h	6-2
Clock prescaler register	E2h	6-2
Memory partition register	E0h	6-1
DMA Registers: Chapter 9		
DMA 1 control register	DAh	9-3
DMA 1 transfer count register	D8h	9-6
DMA 1 destination address high register	D6h	9-7
DMA 1 destination address low register	D4h	9-8
DMA 1 source address high register	D2h	9-9
DMA 1 source address low register	D0h	9-10
DMA 0 control register	CAh	9-3
DMA 0 transfer count register	C8h	9-6
DMA 0 destination address high register	C6h	9-7
DMA 0 destination address low register	C4h	9-8
DMA 0 source address high register	C2h	9-9
DMA 0 source address low register	C0h	9-10
Chip-Select Registers: Chapter 5		
$\overline{PCS}$ and $\overline{MCS}$ auxiliary register	A8h	5-10
Midrange memory chip-select register	A6h	5-8
Peripheral chip-select register	A4h	5-12
Low memory chip-select register	A2h	5-6
Upper memory chip-select register	A0h	5-4
Serial Port 0 Registers: Chapter 10		
Serial port 0 baud rate divisor register	88h	10-13
Serial port 0 receive register	86h	10-12
Serial port 0 transmit register	84h	10-11
Serial port 0 status register	82h	10-9
Serial port 0 control register	80h	10-5
PIO Registers: Chapter 11		
PIO data 1 register	7Ah	11-5
PIO direction 1 register	78h	11-4
PIO mode 1 register	76h	11-3
PIO data 0 register	74h	11-5
PIO direction 0 register	72h	11-4
PIO mode 0 register	70h	11-3
Timer Registers: Chapter 8		
Timer 2 mode/control register	66h	8-5
Timer 2 max count compare A register	62h	8-7
Timer 2 count register	60h	8-6
Timer 1 mode/control register	5Eh	8-3
Timer 1 max count compare B register	5Ch	8-7
Timer 1 max count compare A register	5Ah	8-7
Timer 1 count register	58h	8-6
Timer 0 mode/control register	56h	8-3
Timer 0 max count compare B register	54h	8-7
Timer 0 max count compare A register	52h	8-7
Timer 0 count register	50h	8-6
Interrupt Registers: Chapter 7		
Serial port 0 interrupt control register	44h	7-18
Serial port 1 interrupt control register	42h	7-18
INT4 interrupt control register	40h	7-16
INT3 control register	3Eh	7-15
INT2 control register	3Ch	7-15
INT1 control register	3Ah	7-14
INT0 control register	38h	7-14
DMA1/INT6 interrupt control register	36h	7-17
DMA0/INT5 interrupt control register	34h	7-17
Timer interrupt control register	32h	7-29
Interrupt status register	30h	7-19
Interrupt request register	2Eh	7-20
Interrupt in-service register	2Ch	7-22
Interrupt priority mask register	2Ah	7-23
Interrupt mask register	28h	7-34
Interrupt poll status register	26h	7-25
Interrupt poll register	24h	7-26
End-of-interrupt register	22h	7-27
Interrupt vector register	20h	7-36
Serial Port 1 Registers: Chapter 10		
Serial port 1 baud rate divisor register	18h	10-13
Serial port 1 receive register	16h	10-12
Serial port 1 transmit register	14h	10-11
Serial port 1 status register	12h	10-9
Serial port 1 control register	10h	10-5

Note: All unused addresses are reserved and should not be accessed.

4.1.1 Peripheral Control Block Relocation Register (RELREG, Offset FEh)

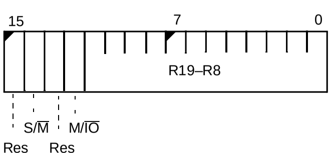
The peripheral control block is mapped into either memory or I/O space by programming the Peripheral Control Block Relocation (RELREG) register (see Figure 4-1). This register is a 16-bit register at offset FEh from the control block base address. The Peripheral Control Block Relocation register provides the upper 12 bits of the base address of the control block. The control block is effectively an internal chip select range.

Other chip selects can overlap the control block only if they are programmed to zero wait states and ignore external ready. If the control register block is mapped into I/O space, the upper four bits of the base address must be programmed as 0000b (since I/O addresses are only 16 bits wide).

In addition to providing relocation information for the control block, the Peripheral Control Block Relocation register contains a bit that places the interrupt controller into either slave mode or master mode.

At reset, the Peripheral Control Block Relocation register is set to 20FFh, which maps the control block to start at FF00h in I/O space. An offset map of the 256-byte peripheral control register block is shown in Table 4-1.

Figure 4-1 Peripheral Control Block Relocation Register



The value of the RELREG register is 20FFh at reset.

Bit 15: Reserved

Bit 14: Slave/Master (S/$\overline{M}$)—Configures the interrupt controller for slave mode when set to 1 and for master mode when set to 0.

Bit 13: Reserved

Bit 12: Memory/IO Space (M/$\overline{IO}$)—When set to 1, the peripheral control block (PCB) is located in memory space. When set to 0, the PCB is located in I/O space.

Bits 11–0: Relocation Address Bits (R19–R8)—R19–R8 define the upper address bits of the PCB base address. The lower eight bits (R7–R0) default to 00h. R19–R16 are ignored when the PCB is mapped to I/O space.



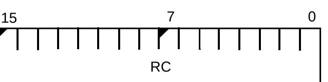
4.1.2 Reset Configuration Register (RESCON, Offset F6h)

The Reset Configuration (RESCON) register (see Figure 4-2) in the peripheral control block latches system-configuration information that is presented to the processor on the address/data bus (AD15–AD0 for the 186 or AO15–AO8 and AD7–AD1 for the 188) during the rising edge of reset. The interpretation of this information is system-specific. The processor does not impose any predetermined interpretation, but simply provides a means for communicating this information to software.

When the $\overline{\text{RES}}$ input is asserted Low, the contents of the address/data bus are written into the Reset Configuration register. The system can place configuration information on the address/data bus using weak external pullup or pulldown resistors, or using an external driver that is enabled during reset. The processor does not drive the address/data bus during reset.

For example, the Reset Configuration register could be used to provide the software with the position of a configuration switch in the system. Using weak external pullup and pulldown resistors on the address and data bus, the system could provide the microcontroller with a value corresponding to the position of a jumper during a reset.

Figure 4-2 Reset Configuration Register



The value of the RESCON register is system-dependent.

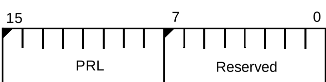
Bits 15–0: Reset Configuration (RC)—There is a one-to-one correspondence between address/data bus signals during the reset and the Reset Configuration register's bits. On the Am186ES microcontroller, AD15 corresponds to bit 15 of the Reset Configuration register, and so on. On the Am188ES microcontroller, AO15 corresponds to register bit 15 and AD7 corresponds to bit 7. Once $\overline{\text{RES}}$ is deasserted, the Reset Configuration register holds its value. This value can be read by software to determine the configuration information.

The contents of the Reset Configuration register are read-only and remain valid until the next processor reset.

4.1.3 Processor Release Level Register (PRL, Offset F4h)

The Processor Release Level register (Figure 4-3) is a read-only register that specifies the processor version.

Figure 4-3 Processor Release Level Register



Bits 15–8: Processor Release Level (PRL)—This byte returns the current release level of the processor, as well as the identification of the family member. The Am186ES and Am188ES microcontrollers' revision A PRL is 10h.

Bits 7–0: Reserved

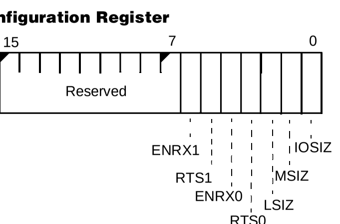
Table 4-2 Processor Release Level (PRL) Values

PRL Value	Processor Release Level
10h	A
11h	B
12h	C
13h	D
14h	E

4.1.4 Auxiliary Configuration Register (AUXCON, Offset F2h)

The auxiliary configuration register is used to configure the asynchronous serial port flow-control signals and to configure the data bus width for memory and I/O accesses. The format of the auxiliary configuration register is shown in Figure 4-4.

Figure 4-4 Auxiliary Configuration Register



The reset value of this register is 0000h.

Bits 15–7: Reserved

Bit 6: Serial Port 1 Enable Receiver Request (ENRX1)—When this bit is 1, the $\overline{CTS1}$/$\overline{ENRX1}$ pin is configured as $\overline{ENRX1}$. When this bit is 0, the $\overline{CTS1}$/$\overline{ENRX1}$ pin is configured as $\overline{CTS1}$. This bit is 0 after processor reset.

Bit 5: Serial Port 1 Request to Send (RTS1)—When this bit is 1, the $\overline{RTR1}$/$\overline{RTS1}$ pin is configured as $\overline{RTS1}$. When this bit is 0, the $\overline{RTR1}$/$\overline{RTS1}$ pin is configured as $\overline{RTR1}$. This bit is 0 after processor reset.

Bit 4: Serial Port 0 Enable Receiver Request (ENRX0)—When this bit is 1, the $\overline{CTS0}$/$\overline{ENRX0}$ pin is configured as $\overline{ENRX0}$. When this bit is 0, the $\overline{CTS0}$/$\overline{ENRX0}$ pin is configured as $\overline{CTS0}$. This bit is 0 after processor reset.

Bit 3: Serial Port 0 Request to Send (RTS0)—When this bit is 1, the $\overline{RTR0}$/$\overline{RTS0}$ pin is configured as $\overline{RTS0}$. When this bit is 0, the $\overline{RTR0}$/$\overline{RTS0}$ pin is configured as $\overline{RTR0}$. This bit is 0 after processor reset.

Bit 2: LCS Data Bus Size (LSIZ)—(Am186ES microcontroller only) This bit determines the width of the data bus for accesses to LCS space. If this bit is 1, 8-bit accesses are performed. If this bit is 0, 16-bit accesses are performed. This bit should not be modified while executing from LCS space or while the PCB is overlaid with LCS space. This bit is 0 after processor reset.

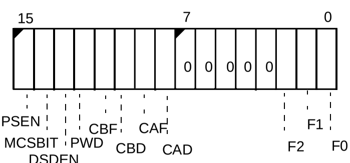
Bit 1: Midrange Data Bus Size (MSIZ)—(Am186ES microcontroller only) This bit determines the width of the data bus for memory accesses which do not fall into the UCS or LCS address spaces, including MCS address space and PCS address space, if mapped to memory. If this bit is 1, 8-bit accesses are performed. If this bit is 0, 16-bit accesses are performed. This bit should not be modified while executing from the associated address space or while the PCB is overlaid on this address space. This bit is 0 after processor reset.

Bit 0: I/O Space Data Bus Size (IOSIZ)—(Am186ES microcontroller only) This bit determines the width of the data bus for all I/O space accesses. If this bit is 1, 8-bit accesses are performed. If this bit is 0, 16-bit accesses are performed. This bit is 0 after processor reset. This bit should not be modified while the PCS is located in I/O space.

4.1.5 System Configuration Register (SYSCON, Offset F0h)

The format of the system configuration register is shown in Figure 4-5.

Figure 4-5 System Configuration Register



The value of the SYSCON register at reset is 0000h.

Bit 15: Enable Power-Save Mode (PSEN)—When set to 1, enables power-save mode and divides the internal operating clock by the value in F2–F0. PSEN is automatically cleared when an external interrupt, including those generated by on-chip peripheral devices, occurs. The value of the PSEN bit is not restored by the execution of an IRET instruction. Software interrupts (INT instruction) and exceptions do not clear the PSEN bit, and interrupt service routines for these conditions should do so, if desired. This bit is 0 after processor reset.

Bit 14: MCS0 Only Mode Bit (MCSBIT)—This bit controls the MCS0 only mode. When set to 0, the middle chip selects operate normally. When set to 1, $\overline{MCS0}$ is active over the entire MCS range. This bit is 0 after processor reset.

Bit 13: Data Strobe Mode of DEN Enable (DSDEN)—This bit enables the data strobe timings on the $\overline{DEN}$ pin. When this bit is set to 1, data strobe bus mode is enabled, and the $\overline{DS}$ timing for reads and writes is identical to the normal read cycle DEN timing. When this bit is set to 0, the $\overline{DEN}$ timing for both reads and writes is normal. The $\overline{DEN}$ pin is renamed $\overline{DS}$ in data strobe bus mode. This bit is 0 after processor reset.

During the bus cycle in which the DSDEN bit of the SYSCON register is written, the timing of the $\overline{DEN}$/$\overline{DS}$ pin is slightly different from normal. When a 1 is written to the DSDEN bit (which previously contained a 0), the falling edge of $\overline{DEN}$/$\overline{DS}$ occurs during PH2 of T_1 as it does during a normal write cycle, but the rising edge occurs during PH1 of T_4 in conformance with the data strobe timing. All writes after this have the normal data strobe timing until the DSDEN bit is reset.

When a 0 is written to the DSDEN bit (which previously contained a 1), the falling edge of $\overline{DEN}$/$\overline{DS}$ occurs during PH2 of T_2 as it does with the data strobe timing, but the rising edge occurs during PH2 of T_4 in conformance with normal write cycle timing. All writes after this have the normal write cycle timing until the DSDEN bit is set again.

Bit 12: Pulse Width Demodulation Mode Enable (PWD)—This bit enables pulse width demodulation mode. When this bit is set to 1, pulse width demodulation is enabled. When this bit is set to 0, pulse width demodulation is disabled. This bit is 0 after processor reset.

Bit 11: CLKOUTB Output Frequency (CBF)—When set to 1, CLKOUTB follows the crystal input (PLL) frequency. When set to 0, CLKOUTB follows the internal processor frequency (after the clock divisor). This bit is 0 after processor reset.



CLKOUTB can be used as a full-speed clock source in power-save mode.

Bit 10: CLKOUTB Drive Disable (CBD)—When set to 1, CBD three-states the clock output driver for CLKOUTB. When set to 0, CLKOUTB is driven as an output. This bit is 0 after processor reset.

Bit 9: CLKOUTA Output Frequency (CAF)—When set to 1, CLKOUTA follows the crystal input (PLL) frequency. When set to 0, CLKOUTA follows the internal processor frequency (after the clock divisor). This bit is 0 after processor reset.

CLKOUTA can be used as a full-speed clock source in power-save mode.

Bit 8: CLKOUTA Drive Disable (CAD)—When set to 1, CAD three-states the clock output driver for CLKOUTA. When set to 0, CLKOUTA is driven as an output. This bit is 0 after processor reset.

Bits 7–3: Reserved—Read back as 0.

Bits 2–0: Clock Divisor Select (F2–F0)—Controls the division factor when Power-Save mode is enabled. F2–-F0 is 000b after processor reset. Allowable values are as follows:

F2	F1	F0	Divider Factor
0	0	0	Divide by 1 (2^0)
0	0	1	Divide by 2 (2^1)
0	1	0	Divide by 4 (2^2)
0	1	1	Divide by 8 (2^3)
1	0	0	Divide by 16 (2^4)
1	0	1	Divide by 32 (2^5)
1	1	0	Divide by 64 (2^6)
1	1	1	Divide by 128 (2^7)

4.2 INITIALIZATION AND PROCESSOR RESET

Processor initialization or startup is accomplished by driving the $\overline{RES}$ input pin Low. $\overline{RES}$ must be Low during power-up to ensure proper device initialization. $\overline{RES}$ forces the Am186ES and Am188ES microcontrollers to terminate all execution and local bus activity. No instruction or bus activity occurs as long as $\overline{RES}$ is active.

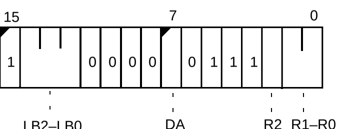
After $\overline{RES}$ is deasserted and an internal processing interval elapses, the microcontroller begins execution with the instruction at physical location FFFF0h. $\overline{RES}$ also sets some registers to predefined values as shown in Table 4-3.

5.5.1 Upper Memory Chip Select Register (UMCS, Offset A0h)

The Am186ES and Am188ES microcontrollers provide the $\overline{UCS}$ chip select pin for the top of memory. On reset, the microcontroller begins fetching and executing instructions starting at memory location FFFF0h, so upper memory is usually used as instruction memory. To facilitate this usage, $\overline{UCS}$ defaults to active on reset with a default memory range of 64 Kbytes from F0000h to FFFFFh, with external ready required and three wait states automatically inserted.

The $\overline{UCS}$ memory range always ends at FFFFFh. The lower boundary is programmable. The Upper Memory Chip Select is configured through the UMCS register (Figure 5-1).

Figure 5-1 Upper Memory Chip Select Register



The value of the UMCS register at reset is F03Bh.

Bit 15: Reserved—Set to 1.

Bits 14–12: Lower Boundary (LB2–LB0)—The LB2–LB0 bits define the lower bound of the memory access through the $\overline{UCS}$ chip selects. The number of programmable bits has been reduced from the eight bits in the 80C186 and 80C188 microcontrollers to three bits in the Am186ES and Am188ES microcontrollers.

The Am186ES and Am188ES microcontrollers provide an additional block size of 512K, which is not available on the 80C186 and 80C188 microcontrollers. Table 5-2 outlines the possible configurations and differences with the 80C186 and 80C188 microcontrollers.

Table 5-2 UMCS Block Size Programming Values

Memory Block Size	Starting Address	LB2–LB0	Comments
64K	F0000h	111b	Default
128K	E0000h	110b	
256K	C0000h	100b	
512K	80000h	000b	Not available on the 80C186 or 80C188 microcontroller



Bits 11–8: Reserved

Bit 7: Disable Address (DA)—The DA bit enables or disables the AD15–AD0 bus during the address phase of a bus cycle when UCS is asserted. If DA is set to 1, AD15–AD0 is not driven during the address phase of a bus cycle when UCS is asserted. If DA is set to 0, AD15–AD0 is driven during the address phase of a bus cycle. Disabling AD15–AD0 reduces power consumption. DA defaults to 0 at power-on reset.

Note: *On the Am188ES microcontroller, the AO15–AO8 address pins are driven during the data phase of the bus cycles, even when DA is set to 1 in either the Upper Memory Chip Select register (UMCS) or the Low Memory Chip Select register (LMCS).*

If BHE/ADEN (on the 186) or RFSH2/ADEN (on the 188) is held Low on the rising edge of RES, then AD15–AD0 is always driven regardless of the DA setting. This configures AD15–AD0 to be enabled regardless of the setting of DA.

If BHE/ADEN (on the 186) or RFSH2/ADEN (on the 188) is High on the rising edge of RES, then DA in the Upper Memory Chip Select (UMCS) register and DA in the Low Memory Chip Select (LMCS) register control the AD15–AD0 disabling.

See the descriptions of the BHE/ADEN and RFSH2/ADEN pins in Chapter 3.

Bit 6: Reserved—Set to 0.

Bits 5–3: Reserved—Set to 1.

Bit 2: Ready Mode (R2)—The R2 bit is used to configure the ready mode for the UCS chip select. If R2 is set to 0, external ready is required. If R2 is set to 1, external ready is ignored. In each case, the processor also uses the value of the R1–R0 bits to determine the number of wait states to insert. R2 defaults to 0 at reset.

Bits 1–0: Wait-State Value (R1–R0)—The value of R1–R0 determines the number of wait states inserted into an access to the UCS memory area. From zero to three wait states can be inserted (R1–R0 = 00b to 11b). R1–R0 default to 11b at reset.

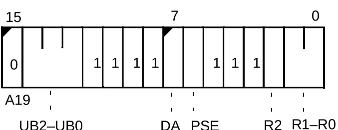
5.5.2 Low Memory Chip Select Register (LMCS, Offset A2h)

The Am186ES and Am188ES microcontrollers provide the LCS chip select pin for the bottom of memory. Since the interrupt vector table is located at 00000h at the bottom of memory, the LCS pin has been provided to facilitate this usage. The LCS pin is not active on reset, but any write access to the LMCS register activates this pin.

Before activating the LCS chip select, the width of the data bus for LCS space should be configured in the AUXCON register.

The Low Memory Chip Select is configured through the LMCS register (see Figure 5-2).

Figure 5-2 Low Memory Chip Select Register



The value of the LMCS register at reset is undefined except DA is set to 0.

Bit 15: Reserved—Set to 0.

Bits 14–12: Upper Boundary (UB2–UB0)—The UB2–UB0 bits define the upper boundary of the memory accessed through the LCS chip select. Because of the timing requirements of the LCS output and the nonmultiplexed address bus, the number of programmable memory sizes for the LMCS register is reduced compared to the 80C186 and 80C188 microcontrollers. Consequently, the number of programmable bits has been reduced from eight bits in the 80C186 and 80C188 microcontrollers to three bits in the Am186ES and Am188ES microcontrollers.

The Am186ES and Am188ES microcontrollers have a block size of 512 Kbytes, which is not available on the 80C186 and 80C188 microcontrollers. Table 5-3 outlines the possible configurations.

Table 5-3 LMCS Block Size Programming Values

Memory Block Size	Ending Address	UB2–UB0
64K	0FFFFh	000b
128K	1FFFFh	001b
256K	3FFFFh	011b
512K	7FFFFh	111b

Bits 11–8: Reserved—Set to 1.

Bit 7: Disable Address (DA)—The DA bit enables or disables the AD15–AD0 bus during the address phase of a bus cycle when LCS is asserted. If DA is set to 1, AD15–AD0 is not driven during the address phase of a bus cycle when LCS is asserted. If DA is set to 0, AD15–AD0 is driven during the address phase of a bus cycle. Disabling AD15–AD0 reduces power consumption. This bit is 0 after processor reset.

Note: *On the Am188ES microcontroller, the AO15–AO8 address pins are driven during the data phase of the bus cycles, even when DA is set to 1 in either the Upper Memory Chip Select register (UMCS) or the Low Memory Chip Select register (LMCS).*

If BHE/ADEN (on the 186) or RFSH2/ADEN (on the 188) is held Low on the rising edge of RES, then AD15–AD0 is always driven regardless of the DA setting.

If BHE/ADEN (on the 186) or RFSH2/ADEN (on the 188) is High on the rising edge of RES, then DA in the UMCS register and DA in the LMCS register control the AD15–AD0 disabling.

See the descriptions of the BHE/ADEN and RFSH2/ADEN pins in Chapter 3.

Bit 6: PSRAM Mode Enable (PSE)—The PSE bit is used to enable PSRAM support for the LCS chip select memory space. When PSE is set to 1, PSRAM support is enabled. When PSE is set to 0, PSRAM support is disabled. The refresh control unit registers, EDRAM, MDRAM, and CDRAM, must be configured for auto refresh before PSRAM support is enabled.

MCS3/RFSH is configured as RFSH by setting the enable bit (EN) in the enable RCU register (EDRAM, offset E4h).

Bits 5–3: Reserved—Set to 1.

Bit 2: Ready Mode (R2)—The R2 bit is used to configure the ready mode for the LCS chip select. If R2 is set to 0, external ready is required. If R2 is set to 1, external ready is ignored. In each case, the processor also uses the value of the R1–R0 bits to determine the number of wait states to insert.

Bits 1–0: Wait-State Value (R1–R0)—The value of R1–R0 determines the number of wait states inserted into an access to the LCS memory area. From zero to three wait states can be inserted (R1–R0 = 00b to 11b).

5.5.3 Midrange Memory Chip Select Register (MMCS, Offset A6h)

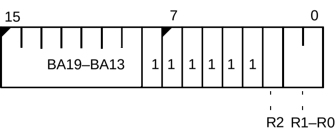
The Am186ES and Am188ES microcontrollers provide four chip select pins, MCS3–MCS0, for use within a user-locatable memory block. The base address of the memory block can be located anywhere within the 1-Mbyte memory address space, exclusive of the areas associated with the UCS and LCS chip selects (and, if they are mapped to memory, the address range of the Peripheral Chip Selects, PCS6–PCS5 and PCS3–PCS0). The MCS address range can overlap the PCS address range if the PCS chip selects are mapped to I/O space.

The Midrange Memory Chip Selects are programmed through two registers. The Midrange Memory Chip Select (MMCS) register (see Figure 5-3) determines the base address and the ready condition and wait states of the memory block accessed through the MCS pins. The PCS and MCS Auxiliary (MPCS) register is used to configure the block size. The MCS3–MCS0 pins are not active on reset. Both the MMCS and MPCS registers must be accessed with a read to activate these chip selects.

Unlike the UCS and LCS chip selects, the MCS3–MCS0 outputs assert with the multiplexed AD address bus (AD15–AD0 or AO15–AO8 and AD7–AD0) rather than the earlier timing of the A19–A0 bus. The A19–A0 bus can still be used for address selection, but the timing is delayed for a half cycle later than that for UCS and LCS.

The Midrange Memory Chip Selects are configured by the MMCS register (Figure 5-3).

Figure 5-3 Midrange Memory Chip Select Register



The value of the MMCS register at reset is undefined.

Bits 15–9: Base Address (BA19–BA13)—The base address of the memory block that is addressed by the MCS chip select pins is determined by the value of BA19–BA13. These bits correspond to bits A19–A13 of the 20-bit memory address. Bits A12–A0 of the base address are always 0.

The base address can be set to any integer multiple of the size of the memory block size selected in the MPCS register. For example, if the midrange block is 32 Kbytes, the block could be located at 10000h or 18000h but not at 14000h.

The base address of the midrange chip selects can be set to 00000h only if the LCS chip select is not active. This is because the LCS base address is defined to be address 00000h and chip select address ranges are not allowed to overlap. Because of the additional restriction that the base address must be a multiple of the block size, a 512K MMCS block size can only be used when located at address 00000h, and the LCS chip selects must not be active in this case. Use of the MCS chip selects to access low memory allows the timing of these accesses to follow the AD address bus rather than the A address bus. Locating a 512K MMCS block at 80000h always conflicts with the range of the UCS chip select and is not allowed.

Bits 8–3: Reserved—Set to 1.

Bit 2: Ready Mode (R2)—The R2 bit is used to configure the ready mode for the MCS chip selects. If R2 is set to 0, external ready is required. If R2 is set to 1, external ready is ignored. In each case, the processor also uses the value of the R1–R0 bits to determine the number of wait states to insert.

Bits 1–0: Wait-State Value (R1–R0)—The value of R1–R0 determines the number of wait states inserted into an access to the MCS memory area. From zero to three wait states can be inserted (R1–R0 = 00b to 11b).

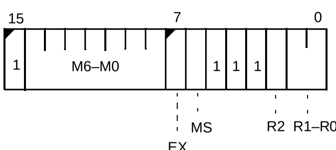
5.5.4 PCS and MCS Auxiliary Register (MPCS, Offset A8h)

The PCS and MCS Auxiliary (MPCS) register (see Figure 5-4) differs from the other chip select control registers in that it contains fields that pertain to more than one type of chip select. The MPCS register fields provide program information for MCS3–MCS0 as well as PCS6–PCS5 and PCS3–PCS0.

In addition to its function as a chip select control register, the MPCS register contains a field that configures the PCS6–PCS5 pins as either chip selects or as alternate sources for the A2 and A1 address bits. When programmed to provide address bits A1 and A2, PCS6–PCS5 cannot be used as peripheral chip selects. These outputs can be used to provide latched address bits for A2 and A1.

On reset, PCS6–PCS5 are not active. If PCS6–PCS5 are configured as address pins, an access to the MPCS register causes the pins to activate. No corresponding access to the PACS register is required to activate the PCS6–PCS5 pins as addresses.

Figure 5-4 PCS and MCS Auxiliary Register



The value of the MPCS register at reset is undefined.

Bit 15: Reserved—Set to 1.

Bits 14–8: MCS Block Size (M6–M0)—This field determines the total block size for the MCS3–MCS0 chip selects. Each individual chip select is active for one quarter of the total block size. The size of the memory block defined is shown in Table 5-4.

Only one of the bits M6–M0 can be set at any time. If more than one of the M6–M0 bits is set, unpredictable operation of the MCS lines occurs.

If the MCSBIT in the SYSCON register is set, MCS0 asserts over the entire programmed block size. MCS3–MCS1 will continue to assert over their programmed range but are typically used as PIOs in the configuration.

Table 5-4 MCS Block Size Programming

Total Block Size	Individual Select Size	M6–M0
8K	2K	0000001b
16K	4K	0000010b
32K	8K	0000100b
64K	16K	0001000b
128K	32K	0010000b
256K	64K	0100000b
512K	128K	1000000b



Bit 7: Pin Selector (EX)—This bit determines whether the $\overline{PCS}$6–$\overline{PCS}$5 pins are configured as chip selects or as alternate outputs for A2–A1. When this bit is set to 1, $\overline{PCS}$6–$\overline{PCS}$5 are configured as peripheral chip select pins. When EX is set to 0, $\overline{PCS}$5 becomes address bit A1 and $\overline{PCS}$6 becomes address bit A2.

Bit 6: Memory/ I/O Space Selector (MS)—This bit determines whether the $\overline{PCS}$ pins are active during memory bus cycles or I/O bus cycles. When MS is set to 1, the $\overline{PCS}$ outputs are active for memory bus cycles. When MS is set to 0, the $\overline{PCS}$ outputs are active for I/O bus cycles.

Bits 5–3: Reserved—Set to 1.

Bit 2: Ready Mode (R2)—This bit applies only to the $\overline{PCS}$6–$\overline{PCS}$5 chip selects. If R2 is set to 0, external ready is required. If R2 is set to 1, external ready is ignored. In each case, the processor also uses the value of the R1–R0 bits to determine the number of wait states to insert.

Bits 1–0: Wait-State Value (R1–R0)—These bits apply only to the $\overline{PCS}$6–$\overline{PCS}$5 chip selects. The value of R1–R0 determines the number of wait states inserted into an access to the $\overline{PCS}$ memory or I/O area. From zero to three wait states can be inserted (R1–R0 = 00b to 11b).

5.5.5 Peripheral Chip Select Register (PACS, Offset A4h)

Unlike the $\overline{UCS}$ and $\overline{LCS}$ chip selects, the $\overline{PCS}$ outputs assert with the same timing as the multiplexed AD address bus. Also, each peripheral chip select asserts over a 256-byte address range, which is twice the address range covered by peripheral chip selects in the 80C186 and 80C188 microcontrollers.

The Am186ES and Am188ES microcontrollers provide six chip selects, $\overline{PCS}$6–$\overline{PCS}$5 and $\overline{PCS}$3–$\overline{PCS}$0, for use within a user-locatable memory or I/O block. ($\overline{PCS}$4 is not implemented on the Am186ES and Am 188ES microcontrollers). The base address of the memory block can be located anywhere within the 1-Mbyte memory address space, exclusive of the areas associated with the $\overline{UCS}$, $\overline{LCS}$, and $\overline{MCS}$ chip selects, or they can be configured to access the 64-Kbyte I/O space.

The Peripheral Chip Selects are programmed through two registers—the Peripheral Chip Select (PACS) register and the $\overline{PCS}$ and $\overline{MCS}$ Auxiliary (MPCS) register. The Peripheral Chip Select (PACS) register (Figure 5-5) determines the base address, the ready condition, and the wait states for the $\overline{PCS}$3–$\overline{PCS}$0 outputs.

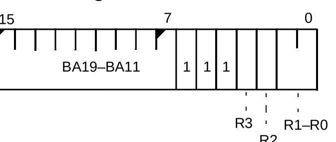
The $\overline{PCS}$ and $\overline{MCS}$ Auxiliary (MPCS) register (see Figure 5-4) contains bits that configure the $\overline{PCS}$6–$\overline{PCS}$5 pins as either chip selects or address pins A1 and A2. When the $\overline{PCS}$6–$\overline{PCS}$5 pins are chip selects, the MPCS register also determines whether $\overline{PCS}$ chip selects are active during memory or I/O bus cycles and specifies the ready and wait states for the $\overline{PCS}$6–$\overline{PCS}$5 outputs.

The $\overline{PCS}$ pins are not active on reset. The PCS pins are activated as chip selects by writing to the PACS and MPCS registers.

$\overline{PCS}$6–$\overline{PCS}$5 can be configured and activated as address pins by writing only the MPCS register. No corresponding access to the PACS register is required in this case.

$\overline{PCS}$3–$\overline{PCS}$0 can be configured for zero wait states to 15 wait states. $\overline{PCS}$6–$\overline{PCS}$5 can be configured for zero wait states to three wait states.

Figure 5-5 Peripheral Chip Select Register



The value of the PACS register at reset is undefined.

Bits 15–7: Base Address (BA19–BA11)—The base address of the peripheral chip select block is defined by BA19–BA11 of the PACS register. BA19–BA11 correspond to bits 19–11 of the 20-bit programmable base address of the peripheral chip select block. Bit 6 of the PACS register corresponds to bit 10 of the base address in the original 80C186 and 80C188 microcontrollers and is not implemented. Thus, code previously written for the 80C186 microcontroller in which bit 6 was set with a meaningful value would not produce the address expected on the Am186ES microcontroller.

When the $\overline{PCS}$ chip selects are mapped to I/O space, BA19–16 must be programmed to 0000b because the I/O address bus is only 16-bits wide.

Table 5-5 PCS Address Ranges

PCS Line	Range	
	Low	High
PCS0	Base Address	Base Address+255
PCS1	Base Address+256	Base Address+511
PCS2	Base Address+512	Base Address+767
PCS3	Base Address+768	Base Address+1023
Reserved	N/A	N/A
PCS5	Base Address+1280	Base Address+1535
PCS6	Base Address+1536	Base Address+1791

Bits 6–4: Reserved—Set to 1.

Bit 3: Wait-State Value (R3)—If this bit is set to 0, the number of wait states from zero to three is encoded in the R1–R0 bits. In this case, R1–R0 encodes from zero (00b) to three (11b) wait states.

When R3 is set to 1, the four possible values of R1–R0 encode four additional wait-state values as follows: 00b = 5 wait states, 01b = 7 wait states, 10b = 9 wait states, and 11b = 15 wait states. Table 5-6 shows the wait-state encoding.

Table 5-6 PCS3–PCS0 Wait-State Encoding

R3	R1	R0	Wait States
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	5
1	0	1	7
1	1	0	9
1	1	1	15

Bit 2: Ready Mode (R2)—The R2 bit is used to configure the ready mode for the PCS3–PCS0 chip selects. If R2 is set to 0, external ready is required. External ready is ignored when R2 is set to 1. In each case, the processor also uses the value of the R3 and R1–R0 bits to determine the number of wait states to insert. The ready mode for PCS6–PCS5 is configured through the MPCS register.

Bits 1–0: Wait-State Value (R1–R0)—The value of R3 and R1–R0 determines the number of wait states inserted into a PCS3–PCS0 access. Up to 15 wait states can be inserted.

See the discussion of bit 3 (R3) for the wait-state encoding of R1–R0.

From zero to three wait states for the PCS6–PCS5 outputs are programmed through the R1–R0 bits in the MPCS register.

Table 7-1 Am186ES and Am188ES Microcontroller Interrupt Types

Interrupt Name	Interrupt Type	Vector Table Address	EOI Type	Overall Priority	Related Instructions	Notes
Divide Error Exception	00h	00h	N/A	1	DIV, IDIV	1
Trace Interrupt	01h	04h	N/A	1A	All	2
Nonmaskable Interrupt (NMI)	02h	08h	N/A	1B		
Breakpoint Interrupt	03h	0Ch	N/A	1	INT3	1
INTO Detected Overflow Exception	04h	10h	N/A	1	INTO	1
Array Bounds Exception	05h	14h	N/A	1	BOUND	1
Unused Opcode Exception	06h	18h	N/A	1	Undefined Opcodes	1
ESC Opcode Exception	07h	1Ch	N/A	1	ESC Opcodes	1, 3
Timer 0 Interrupt	08h	20h	08h	2A		4, 5
Timer 1 Interrupt	12h	48h	08h	2B		4, 5
Timer 2 Interrupt	13h	4Ch	08h	2C		4, 5
Reserved for AMD Use	09h	24h				
DMA 0 Interrupt/INT5	0Ah	28h	0Ah	3		5
DMA 1 Interrupt/INT6	0Bh	2Ch	0Bh	4		5
INT0 Interrupt	0Ch	30h	0Ch	5		
INT1 Interrupt	0Dh	34h	0Dh	6		
INT2 Interrupt	0Eh	38h	0Eh	7		
INT3 Interrupt	0Fh	3Ch	0Fh	8		
INT4 Interrupt	10h	40h	10h	9		6
Asynchronous Serial Port 1 Interface	11h	42h	11h	9		6
Asynchronous Serial Port 0 Interrupt	14h	44h	14h	9		6
Reserved for AMD Use	15h–1Fh	54h–7Ch				

Notes:

Default priorities for the interrupt sources are used if the user does not reprogram priority levels.

1. *Interrupts generate as a result of an instruction execution.*
2. *Trace is performed in the same manner as 8086 and 8088.*
3. *An ESC opcode causes a trap.*
4. *All three timers constitute one source of request to the interrupt controller. As such, they share the same priority level with respect to other interrupt sources. However, the timers have a defined priority order among themselves (2A>2B>2C).*
5. *The interrupt types of these sources are programmable in slave mode.*
6. *Not available in slave mode.*



7.3 MASTER MODE INTERRUPT CONTROLLER REGISTERS

The interrupt controller registers for master mode are shown in Table 7-2. All the registers can be read and written unless otherwise specified.

Registers can be redefined in slave mode. See Section 7.4 on page 7-28 for detailed information regarding slave mode register usage. On reset, the microcontroller is in master mode. Bit 14 of the Peripheral Control Block Relocation register (see Figure 4-1) must be set to initiate slave mode operation.

Table 7-2 Interrupt Controller Registers in Master Mode

Offset	Register Mnemonic	Register Name	Associated Pins	Comments
38h	I0CON	INT0 Control	INT0	
3Ah	I1CON	INT1 Control	INT1	
3Ch	I2CON	INT2 Control	INT2	
3Eh	I3CON	INT3 Control	INT3	
40h	I4CON	INT4 Control	INT4	
34h	DMA0CON	DMA0 Interrupt Control/INT5	INT5	
36h	DMA1CON	DMA1 Interrupt Control/INT6	INT6	
32h	TCUCON	Timer Interrupt Control	TMRIN1 TMRIN0 TMROUT1 TMROUT0	
44h	SP0CON	Serial Port 0 Interrupt Control		
42h	SP1CON	Serial Port 1 Interrupt Control		
30h	INTSTS	Interrupt Status		
2Eh	REQST	Interrupt Request	INT6–INT0	Read-only register
2Ch	INSERV	In-Service	INT6–INT0	
2Ah	PRIMSK	Priority Mask		
28h	IMASK	Interrupt Mask	INT6–INT0	
26h	POLLST	Poll Status		Read-only register
24h	POLL	Poll		Read-only register
22h	EOI	End of Interrupt		Write-only register

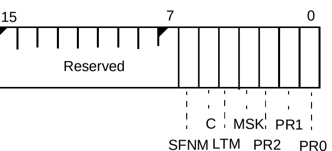
7.3.1 INT0 and INT1 Control Registers (I0CON, Offset 38h, I1CON, Offset 3Ah) (Master Mode)

The INT0 interrupt is assigned to interrupt type 0Ch. The INT1 interrupt is assigned to interrupt type 0Dh.

When cascade mode is enabled for INT0 by setting the C bit of I0CON to 1, the INT2 pin becomes INTA0, the interrupt acknowledge for INT0.

When cascade mode is enabled for INT1 by setting the C bit of I1CON to 1, the INT3 pin becomes INTA1, the interrupt acknowledge for INT1.

Figure 7-4 INT0 and INT1 Control Registers



The value of I0CON and I1CON at reset is 000Fh.

Bits 15–7: Reserved—Set to 0.

Bit 6: Special Fully Nested Mode (SFNM)—When set to 1, enables special fully nested mode for INT0 or INT1.

Bit 5: Cascade Mode (C)—When set to 1, this bit enables cascade mode for INT0 or INT1.

Bit 4: Level-Triggered Mode (LTM)—This bit determines whether the microcontroller interprets an INT0 or INT1 interrupt request as edge- or level-sensitive. A 1 in this bit configures INT0 or INT1 as an active High, level-sensitive interrupt. A 0 in this bit configures INT0 or INT1 as a Low-to-High, edge-triggered interrupt. In either case, INT0 or INT1 must remain High until they are acknowledged.

Bit 3: Mask (MSK)—This bit determines whether the INT0 or INT1 signal can cause an interrupt. A 1 in this bit masks this interrupt source, preventing INT0 or INT1 from causing an interrupt. A 0 in this bit enables INT0 or INT1 interrupts.

This bit is duplicated in the Interrupt Mask register. See the Interrupt Mask register in Section 7.3.10 on page 7-24.

Bits 2–0: Priority Level (PR2–PR0)—This field determines the priority of INT0 or INT1 relative to the other interrupt signals, as shown in Table 7-3 on page 7-18.

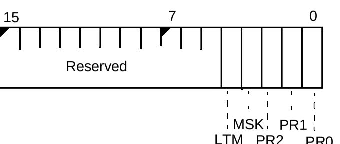


7.3.2 INT2 and INT3 Control Registers (I2CON, Offset 3Ch, I3CON, Offset 3Eh) (Master Mode)

The INT2 interrupt is assigned to interrupt type OEh. The INT3 interrupt is assigned to interrupt type 0Fh.

The INT2 and INT3 pins can be configured as interrupt acknowledge pins INTA0 and INTA1 when cascade mode is implemented.

Figure 7-5 INT2 and INT3 Control Registers



The value of I2CON and I3CON at reset is 000Fh.

Bits 15–5: Reserved—Set to 0.

Bit 4: Level-Triggered Mode (LTM)—This bit determines whether the microcontroller interprets an INT2 or INT3 interrupt request as edge- or level-sensitive. A 1 in this bit configures INT2 or INT3 as an active High, level-sensitive interrupt. A 0 in this bit configures INT2 or INT3 as a Low-to-High, edge-triggered interrupt. In either case, INT2 or INT3 must remain High until they are acknowledged.

Bit 3: Mask (MSK)—This bit determines whether the INT2 or INT3 signal can cause an interrupt. A 1 in this bit masks this interrupt source, preventing INT2 or INT3 from causing an interrupt. A 0 in this bit enables INT2 or INT3 interrupts.

This bit is duplicated in the Interrupt Mask register. See the Interrupt Mask register in Section 7.3.10 on page 7-24.

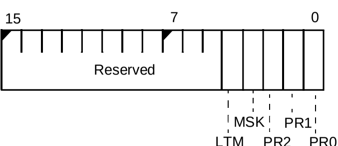
Bits 2–0: Priority Level (PR2–PR0)—This field determines the priority of INT2 or INT3 relative to the other interrupt signals, as shown in Table 7-3 on page 7-18.

7.3.3 INT4 Control Register (I4CON, Offset 40h) (Master Mode)

The Am186ES and Am188ES microcontrollers provide INT4, an additional external interrupt pin. This input behaves like INT3–INT0 on the 80C186 microcontroller with the exception that INT4 is only intended for use as a fully nested-mode interrupt source. INT4 is not available in cascade mode.

This interrupt is assigned to interrupt type 10h. The Interrupt 4 Control register (see Figure 7-6) controls the operation of the INT4 signal.

Figure 7-6 INT4 Control Register



The value of I4CON at reset is 000Fh.

Bits 15–5: Reserved—Set to 0.

Bit 4: Level-Triggered Mode (LTM)—This bit determines whether the microcontroller interprets an INT4 interrupt request as edge- or level-sensitive. A 1 in this bit configures INT4 as an active High, level-sensitive interrupt. A 0 in this bit configures INT4 as a Low-to-High, edge-triggered interrupt. In either case, INT4 must remain High until it is acknowledged.

Bit 3: Mask (MSK)—This bit determines whether the INT4 signal can cause an interrupt. A 1 in this bit masks this interrupt source, preventing INT4 from causing an interrupt. A 0 in this bit enables INT4 interrupts.

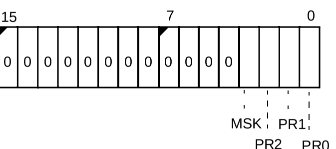
This bit is duplicated in the Interrupt Mask register. See the Interrupt Mask register in Section 7.3.10 on page 7-24.

Bits 2–0: Priority (PR)—This field determines the priority of INT4 relative to the other interrupt signals, as shown in Table 7-3 on page 7-18.

7.3.4 Timer and DMA Interrupt Control Registers (TCUCON, Offset 32h, DMA0CON/INT5CON, Offset 34h, DMA1CON/INT6CON, Offset 36h) (Master Mode)

The three timer interrupts are assigned to interrupt type 08h, 12h, and 13h. All three timer interrupts are configured through TCUCON, offset 32h. The DMA0 interrupt is assigned to interrupt type 0Ah. The DMA1 interrupt is assigned to interrupt type 0Bh. See the DMA control registers for how to configure these pins as DMA requests or external interrupts.

Figure 7-7 Timer/DMA Interrupt Control Registers



The value of TCUCON, DMA0CON, and DMA1CON at reset is 000Fh.

Bits 15–4: Reserved—Set to 0.

Bit 3: Interrupt Mask (MSK)—This bit determines whether the corresponding signal can generate an interrupt. A 1 masks this interrupt source. A 0 enables the corresponding interrupt.

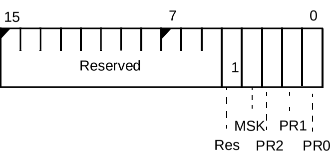
This bit is duplicated in the Interrupt Mask register. See the Interrupt Mask register in Section 7.3.10 on page 7-24.

Bits 2–0: Priority Level (PR2–PR0)—Sets the priority level for its corresponding source. See Table 7-3 on page 7-18.

7.3.5 Serial Port 0/1 Interrupt Control Registers (SP0CON/SP1CON, Offset 44h/42h) (Master Mode)

The serial port interrupt control registers control the operation of the serial ports' interrupt source (SP1 and SP0, bits 10–9 in the interrupt request register). Serial port 0 is assigned to interrupt type 14h and serial port 1 is assigned to interrupt type 11h. The control register format is shown in Figure 7-8.

Figure 7-8 Serial Port 0/1 Interrupt Control Register



The value of SP0CON and SPICON at reset is 001Fh.

Bits 15–5: Reserved—Set to 0.

Bit 4: Reserved—Set to 1.

Bit 3: Mask (MSK)—This bit determines whether the serial port can cause an interrupt. A 1 in this bit masks this interrupt source, preventing the serial port from causing an interrupt. A 0 in this bit enables serial port interrupts.

This bit is duplicated in the Interrupt Mask register. See the Interrupt Mask register in Section 7.3.10 on page 7-24.

Bits 2–0: Priority (PR2–PR0)—This field determines the priority of the serial port relative to the other interrupt signals. After a reset, the priority is 7. See Table 7-3.

Table 7-3 Priority Level

Priority	PR2–PR0
(High) 0	0 0 0b
1	0 0 1b
2	0 1 0b
3	0 1 1b
4	1 0 0b
5	1 0 1b
6	1 1 0b
(Low) 7	1 1 1b

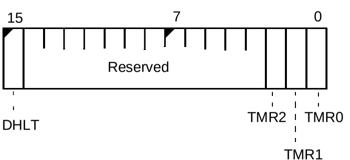


AMD

7.3.6 Interrupt Status Register (INTSTS, Offset 30h) (Master Mode)

The interrupt status register indicates the interrupt request status of the three timers.

Figure 7-9 Interrupt Status Register



Bit 15: DMA Halt (DHLT)—When set to 1, halts any DMA activity. This bit is automatically set to 1 when nonmaskable interrupts occur and is reset when an IRET instruction is executed. Time critical software, such as interrupt handlers, can modify this bit directly to inhibit DMA transfers. Because of the function of this register as an interrupt request register for the timers, the DHLT bit should not be modified by software when timer interrupts are enabled.

Bits 14–3: Reserved

Bits 2–0: Timer Interrupt Request (TMR2–TMR0)—When set to 1, these bits indicate that the corresponding timer has an interrupt request pending. (Note that the timer TMR bit in the REQST register is the logical OR of these timer interrupt requests.)

AMD

7.3.7 Interrupt Request Register (REQST, Offset 2Eh) (Master Mode)

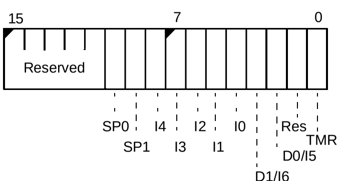
The hardware interrupt sources have interrupt request bits inside the interrupt controller. A read from this register yields the status of these bits. The Interrupt Request register is a read-only register. The format of the Interrupt Request register is shown in Figure 7-10.

For internal interrupts (SP0, SP1, D1/I6, D0/I5, and TMR), the corresponding bit is set to 1 when the device requests an interrupt. The bit is reset during the internally generated interrupt acknowledge.

For INT6–INT0 external interrupts, the corresponding bit (INT4–INT0) reflects the current value of the external signal. The device must hold this signal High until the interrupt is serviced.

Generally the interrupt service routine signals the external device to remove the interrupt request.

Figure 7-10 Interrupt Request Register



The REQST register is undefined on reset.

Bits 15–11: Reserved

Bit 10: Serial Port 0 Interrupt Request (SP0)—This bit indicates the interrupt state of serial port 0. If enabled, the SP0 bit is the logical OR of all possible serial port interrupt sources (THRE, RDR, BRK1, BRK0, FER, PER, and OER status bits).

Bit 9: Serial Port 1 Interrupt Request (SP1)—This bit indicates the interrupt state of serial port 1. If enabled, the SP1 bit is the logical OR of all possible serial port interrupt sources (THRE, RDR, BRK1, BRK0, FER, PER, and OER status bits).

Bits 8–4: Interrupt Requests (INT4–INT0)—When set to 1, the corresponding INT pin has an interrupt pending (i.e., when INT0 is pending, INT0 is set).

Bit 3: DMA Channel 1/Interrupt 6 Request (D1/I6)—When set to 1, DMA channel 1 or INT6 has an interrupt pending.

Bit 2: DMA Channel 0/Interrupt 5 Request (D0/I5)—When set to 1, DMA channel 0 or INT5 has an interrupt pending.

Bit 1: Reserved

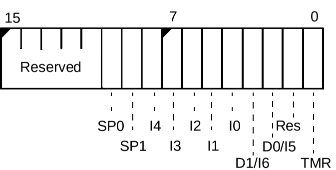
Bit 0: Timer Interrupt Request (TMR)—This bit indicates the state of the timer interrupts. This bit is the logical OR of the timer interrupt requests. When set to a 1, this bit indicates that the timer control unit has an interrupt pending.

The interrupt status register indicates the specific timer that is requesting an interrupt. See Section 7.3.6.

7.3.8 Interrupt In-Service Register (INSERV, Offset 2Ch) (Master Mode)

The bits in the In-Service register are set by the interrupt controller when the interrupt is taken. Each bit in the register is cleared by writing the corresponding interrupt type to the End-of-Interrupt (EOI) register.

Figure 7-11 Interrupt In-Service Register



The INSERV register is set to 0000h on reset.

Bits 15–11: Reserved

Bit 10: Serial Port 0 Interrupt In-Service (SP0)—This bit indicates the in-service state of serial port 0.

Bit 9: Serial Port 1 Interrupt In-Service (SP1)—This bit indicates the in-service state of the serial port 1.

Bits 8–4: Interrupt In-Service (INT4–INT0)—These bits indicate the in-service state of the corresponding INT pin.

Bit 3: DMA Channel 1/Interrupt 6 In-Service (D1/I6)—This bit indicates the in-service state of DMA channel 1 or INT6.

Bit 2: DMA Channel 0/Interrupt 5 In-Service (D0/I5)—This bit indicates the in-service state of DMA channel 0 or INT5.

Bit 1: Reserved

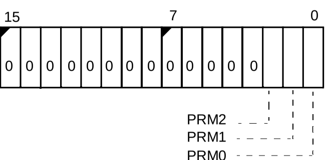
Bit 0: Timer Interrupt In-Service (TMR)—This bit indicates the state of the in-service timer interrupts. This bit is the logical OR of all the timer interrupt requests. When set to a 1, this bit indicates that the corresponding timer interrupt request is in-service.



7.3.9 Priority Mask Register (PRIMSK, Offset 2Ah) (Master Mode)

The Priority Mask register provides the value that determines the minimum priority level at which maskable interrupts can generate an interrupt.

Figure 7-12 Priority Mask Register



The value of PRIMSK at reset is 0007h.

Bits 15–3: Reserved—Set to 0.

Bits 2–0: Priority Field Mask (PRM2–PRM0)—This field determines the minimum priority that is required for a maskable interrupt source to generate an interrupt. Maskable interrupts with programmable priority values that are numerically higher than this field are masked. The possible values are zero (000b) to seven (111b).

A value of seven (111b) allows all interrupt sources that are not masked to generate interrupts. A value of five (101b) allows only unmasked interrupt sources with a programmable priority of zero to five (000b to 101b) to generate interrupts.

Table 7-4 Priority Level

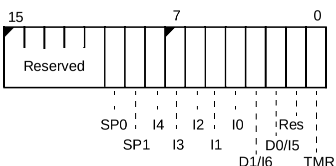
Priority	PR2–PR0
(High) 0	0 0 0b
1	0 0 1b
2	0 1 0b
3	0 1 1b
4	1 0 0b
5	1 0 1b
6	1 1 0b
(Low) 7	1 1 1b

7.3.10 Interrupt Mask Register (IMASK, Offset 28h) (Master Mode)

The Interrupt Mask register is a read/write register. Programming a bit in the Interrupt Mask register has the effect of programming the MSK bit in the associated interrupt control register. The format of the Interrupt Mask register is shown in Figure 7-13.

When a bit is set to 1 in this register, the corresponding interrupt source is masked off. When the bit is set to 0, the interrupt source is enabled to generate an interrupt request.

Figure 7-13 Interrupt Mask Register



The IMASK register is set to 07FDh on reset.

Bits 15–11: Reserved

Bit 10: Serial Port 0 Interrupt Mask (SP0)— When set to 1, this bit indicates that the serial port 0 interrupt is masked.

Bit 9: Serial Port 1 Interrupt Mask (SP1)—When set to 1, this bit indicates that the serial port 1 interrupt is masked.

Bits 8–4: Interrupt Mask (INT4–INT0)—When set to 1, an INT4–INT0 bit indicates that the corresponding interrupt is masked.

Bits 3–2: DMA Channel Interrupt Masks (D1/I6–D0/I5)—When set to 1, a D1/I6–D0/I5 bit indicates that the corresponding DMA or INT6/INT5 channel interrupt is masked.

Bit 1: Reserved

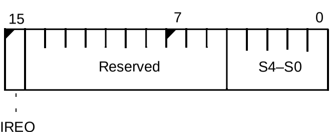
Bit 0: Timer Interrupt Mask (TMR)—When set to 1, this bit indicates that interrupt requests from the timer control unit are masked.



7.3.11 Poll Status Register (POLLST, Offset 26h) (Master Mode)

The Poll Status register mirrors the current state of the Poll register. The Poll Status register can be read without affecting the current interrupt request. But when the Poll register is read, the current interrupt is acknowledged and the next interrupt takes its place in the Poll register. This is a read-only register.

Figure 7-14 Poll Status Register



Bit 15: Interrupt Request (IREQ)—Set to 1 if an interrupt is pending. When this bit is set to 1, the S4–S0 field contains valid data.

Bits 14–5: Reserved—Set to 0.

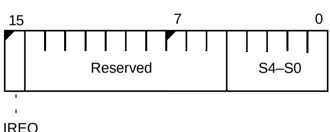
Bits 4–0: Poll Status (S4–S0)—Indicates the interrupt type of the highest priority pending interrupt (see Table 7-1 on page 7-4).

7.3.12 Poll Register (POLL, Offset 24h) (Master Mode)

When the Poll register is read, the current interrupt is acknowledged and the next interrupt takes its place in the Poll register.

The Poll Status register mirrors the current state of the Poll register, but the Poll Status register can be read without affecting the current interrupt request. This is a read-only register.

Figure 7-15 Poll Register



Bit 15: Interrupt Request (IREQ)—Set to 1 if an interrupt is pending. When this bit is set to 1, the S4–S0 field contains valid data.

Bits 14–5: Reserved—Set to 0.

Bits 4–0: Poll Status (S4–S0)—Indicates the interrupt type of the highest priority pending interrupt (see Table 7-1). Reading the Poll register acknowledges the highest pending interrupt and allows the next interrupt to advance into the register.

Although the IS bit is set, the interrupt service routine does not begin execution automatically. The application software must execute the appropriate ISR.

7.3.13 End-of-Interrupt Register (EOI, Offset 22h) (Master Mode)

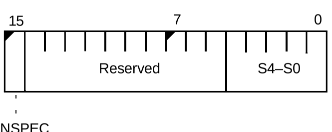
The End-of-Interrupt (EOI) register is a write-only register. The in-service flags in the In-Service register (see Section 7.3.8 on page 7-22) are reset by writing to the EOI register. Before executing the IRET instruction that ends an interrupt service routine (ISR), the ISR should write to the EOI register to reset the IS bit for the interrupt.

The specific EOI reset is the most secure method to use for resetting IS bits. Figure 7-16 shows example code for a specific EOI reset. See Table 7-1 on page 7-4 for specific EOI values.

Figure 7-16 Example EOI Assembly Code

```
           ...
           ...                    ;ISR code
           ...
exit:      mov ax,int_type        ;load the interrupt type in ax
           mov dx, 0ff22h         ;write the interrupt type to EOI
           out dx,ax
           popa
           iret                   ;return from interrupt
```

Figure 7-17 End-of-Interrupt Register



Bit 15: Non-Specific EOI (NSPEC)—The NSPEC bit determines the type of EOI command. When written as a 1, NSPEC indicates non-specific EOI. When written as a 0, NSPEC indicates the specific EOI interrupt type is in S4–S0.

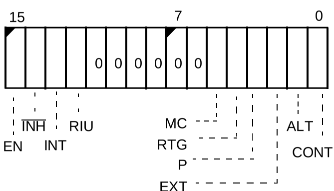
Bits 14–5: Reserved

Bits 4–0: Source Interrupt Type (S4–S0)—Specifies the EOI type of the interrupt that is currently being processed. See Table 7-1 on page 7-4.

8.3.2 Timer 0 and Timer 1 Mode and Control Registers (T0CON, Offset 56h, T1CON, Offset 5Eh)

These registers control the functionality of timer 0 and timer 1. See Figure 8-2.

Figure 8-2 Timer 0 and Timer 1 Mode and Control Registers



The value of T0CON and T1CON at reset is 0000h.

Bit 15: Enable Bit (EN)—When set to 1, the timer is enabled. When set to 0, the timer is inhibited from counting. This bit can only be written with the $\overline{INH}$ bit set at the same time.

Bit 14: Inhibit Bit ($\overline{INH}$)—Allows selective updating of enable (EN) bit. When set to 1 during a write, EN can also be modified. When set to 0 during a write, writes to EN are ignored. This bit is not stored and is always read as 0.

Bit 13: Interrupt Bit (INT)—When set to 1, an interrupt request is generated when the count register equals a maximum count. If the timer is configured in dual maxcount mode, an interrupt is generated each time the count reaches maxcount A or maxcount B. When INT is set to 0, the timer will not issue interrupt requests. If the enable bit is cleared after an interrupt request has been generated but before the pending interrupt is serviced, the interrupt request will still be present.

Bit 12: Register in Use Bit (RIU)—When the maxcount compare A register is being used for comparison to the timer count value, this bit is set to 0. When the maxcount compare B register is being used, this bit is set to 1.

Bits 11–6: Reserved—Set to 0.

Bit 5: Maximum Count Bit (MC)—The MC bit is set to 1 when the timer reaches a maximum count. In dual maxcount mode, the bit is set each time either maxcount compare A or B register is reached. This bit is set regardless of the timer interrupt-enable bit. The MC bit can be used to monitor timer status through software polling instead of through interrupts.

Bit 4: Retrigger Bit (RTG)—Determines the control function provided by the timer input pin. When set to 1, a 0 to 1 edge transition on TMRIN0 or TMRIN1 resets the count. When set to 0, a High input enables counting and a Low input holds the timer value. This bit is ignored when external clocking (EXT=1) is selected.

Bit 3: Prescaler Bit (P)—When set to 1, the timer is prescaled by timer 2. When set to 0, the timer counts up every fourth CLKOUT period. This bit is ignored when external clocking is enabled (EXT=1).

Bit 2: External Clock Bit (EXT)—When set to 1, an external clock is used. When set to 0, the internal clock is used.

Bit 1: Alternate Compare Bit (ALT)—When set to 1, the timer counts to maxcount compare A, then resets the count register to 0. Then the timer counts to maxcount compare B, resets the count register to zero, and starts over with maxcount compare A.

If ALT is clear, the timer counts to maxcount compare A and then resets the count register to zero and starts counting again against maxcount compare A. In this case, mlaxcount compare B is not used.

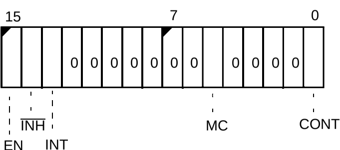
Bit 0: Continuous Mode Bit (CONT)—When set to 1, CONT causes the associated timer to run in the normal continuous mode.

When CONT is set to 0, EN is cleared after each timer count sequence and the timer clears and then halts on reaching the maximum count. If CONT=0 and ALT=1, the timer counts to the maxcount compare A register value and resets, then it counts to the B register value and resets and halts.

8.3.3 Timer 2 Mode and Control Register (T2CON, Offset 66h)

This register controls the functionality of timer 2. See Figure 8-3.

Figure 8-3 Timer 2 Mode and Control Register



The value of T2CON at reset is 0000h.

Bit 15: Enable Bit (EN)—When EN is set to 1, the timer is enabled. When set to 0, the timer is inhibited from counting. This bit cannot be written to unless the $\overline{INH}$ bit is set to 1 during the same write.

Bit 14: Inhibit Bit ($\overline{INH}$)—Allows selective updating of enable (EN) bit. When INH is set to 1 during a write, EN can be modified on the same write. When INH is set to 0 during a write, writes to EN are ignored. This bit is not stored and is always read as 0.

Bit 13: Interrupt Bit (INT)—When INT is set to 1, an interrupt request is generated when the count register equals a maximum count. When INT is set to 0, the timer will not issue interrupt requests. If the EN enable bit is cleared after an interrupt request has been generated but before the pending interrupt is serviced, the interrupt request remains active.

Bits 12–6: Reserved—Set to 0.

Bit 5: Maximum Count Bit (MC)—The MC bit is set to 1 when the timer reaches its maximum count. This bit is set regardless of the timer interrupt-enable bit. The MC bit can be used to monitor timer status through software polling instead of through interrupts.

Bits 4–1: Reserved—Set to 0.

Bit 0: Continuous Mode Bit (CONT)—When CONT is set to 1, it causes the associated timer to run continuously. When set to 0, EN is cleared after each timer count sequence and the timer halts on reaching the maximum count.

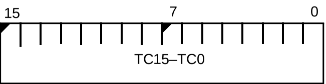


8.3.4 Timer Count Registers (T0CNT, Offset 50h, T1CNT, Offset 58h, T2CNT, Offset 60h)

These registers can be incremented by one every four internal processor clocks. Timer 0 and timer 1 can also be configured to increment based on the TMRIN0 and TMRIN1 external signals, or they can be prescaled by timer 2. See Figure 8-4.

The count registers are compared to maximum count registers and various actions are triggered based on reaching a maximum count.

Figure 8-4 Timer Count Registers



The value of these registers at reset is undefined.

Bits 15–0: Timer Count Value (TC15–TC0)—This register contains the current count of the associated timer. The count is incremented every fourth processor clock in internal clocked mode, or each time the timer 2 maxcount is reached if prescaled by timer 2. Timer 0 and timer 1 can be configured for external clocking based on the TMRIN0 and TMRIN1 signals.

8.3.5 Timer Maxcount Compare Registers (T0CMPA, Offset 52h, T0CMPB, Offset 54h, T1CMPA, Offset 5Ah, T1CMPB, Offset 5Ch, T2CMPA, Offset 62h)

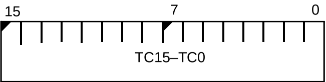
These registers serve as comparators for their associated count registers. Timer 0 and timer 1 each have two maximum count compare registers. See Figure 8-5.

Timer 0 and timer 1 can be configured to count and compare to register A and then count and compare to register B. Using this method, the TMROUT0 or TMROUT1 signals can be used to generate wave forms of various duty cycles.

Timer 2 has one compare register, T2CMPA.

If a maximum count compare register is set to 0000h, the timer associated with that compare register will count from 0000h to FFFFh before requesting an interrupt. With a 40-MHz clock, a timer configured this way interrupts every 6.5536 ms.

Figure 8-5 Timer Maxcount Compare Registers



The value of these registers at reset is undefined.

Bits 15–0: Timer Compare Value (TC15–TC0)—This register contains the maximum value a timer will count to before resetting its count register to 0.

9.3 PROGRAMMABLE DMA REGISTERS

The following sections describe the control registers that are used to configure and operate the two DMA channels.

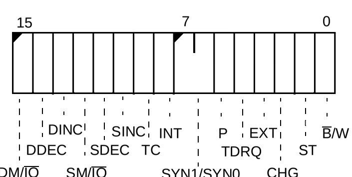
9.3.1 DMA Control Registers (D0CON, Offset CAh, D1CON, Offset DAh)

The DMA control registers (see Figure 9-2) determine the mode of operation for the DMA channels. These registers specify the following options:

- Whether the destination address is memory or I/O space
- Whether the destination address is incremented, decremented, or maintained constant after each transfer
- Whether the source address is memory or I/O space
- Whether the source address is incremented, decremented, or maintained constant after each transfer
- If DMA activity ceases after a programmed number of DMA cycles
- If an interrupt is generated with the last transfer
- The mode of synchronization
- The relative priority of the DMA channel with respect to the other DMA channel
- Whether timer 2 DMA requests are enabled or disabled
- Whether bytes or words are transferred (on the Am186 microcontroller only)
- Whether the DRQ pin is used for external interrupts

The DMA channel control registers can be changed while the channel is operating. Any changes made during DMA operations affect the current DMA transfer.

Figure 9-2 DMA Control Registers



The value of D0CON and D1CON at reset is undefined except ST is set to 0.

Bit 15: Destination Address Space Select (DM/IO)—Selects memory or I/O space for the destination address. When DM/IO is set to 1, the destination address is in memory space. When set to 0, the destination address is in I/O space.

Bit 14: Destination Decrement (DDEC)—When DDEC is set to 1, the destination address is automatically decremented after each transfer. The address decrements by 1 or 2 depending on the byte/word bit (B/W, bit 0). The address remains constant if the increment and decrement bits are set to the same value (00b or 11b).

Bit 13: Destination Increment (DINC)—When DINC is set to 1, the destination address is automatically incremented after each transfer. The address increments by 1 or 2 depending on the byte/word bit (B/W, bit 0). The address remains constant if the increment and decrement bits are set to the same value (00b or 11b).

Bit 12: Source Address Space Select (SM/IO)—When SM/IO is set to 1, the source address is in memory space. When set to 0, the source address is in I/O space.

Bit 11: Source Decrement (SDEC)—When SDEC is set to 1, the source address is automatically decremented after each transfer. The address decrements by 1 or 2, depending on the byte/word bit (B/W, bit 0). The address remains constant if the increment and decrement bits are set to the same value (00b or 11b).

Bit 10: Source Increment (SINC)—When SINC is set to 1, the source address is automatically incremented after each transfer. The address increments by 1 or 2 depending on the byte/word bit (B/W, bit 0). The address remains constant if the increment and decrement bits are set to the same value (00b or 11b).

Bit 9: Terminal Count (TC)—The DMA decrements the transfer count for each DMA transfer. When TC is set to 1, source or destination synchronized DMA transfers terminate when the count reaches 0. When TC is set to 0, source or destination synchronized DMA transfers do not terminate when the count reaches 0. Unsynchronized DMA transfers always terminate when the count reaches 0, regardless of the setting of this bit.

Bit 8: Interrupt (INT)—When INT is set to 1, the DMA channel generates an interrupt request on completion of the transfer count. The TC bit must also be set to generate an interrupt.

Bits 7–6: Synchronization Type (SYN1–SYN0)—The SYN1–SYN0 bits select channel synchronization as shown in Table 9-2. The value of this field is ignored if TDRQ (bit 4) is set to 1. For more information on DMA synchronization, see Section 9.4 on page 9-11. This field is 11b after processor reset.

Table 9-2 Synchronization Type

SYN1	SYN0	Sync Type
0	0	Unsynchronized
0	1	Source Synch
1	0	Destination Synch
1	1	Reserved

Bit 5: Relative Priority (P)—When P is set to 1, it selects high priority for this channel relative to the other channel during simultaneous transfers.

Bit 4: Timer 2 Synchronization (TDRQ)—When TDRQ is set to 1, it enables DMA requests from timer 2. When set to 0, TDRQ disables DMA requests from timer 2.

Bit 3: External Interrupt Enable Bit (EXT)—This bit enables the external interrupt functionality of the corresponding DRQ pin. If this bit is set to 1, the external pin is an INT pin and requests on the pin are processed by the interrupt controller; the associated DMA channel does not respond to changes on the DRQ pin. When this bit is set to 0, the pin functions as a DRQ pin.

Bit 2: Change Start Bit (CHG)—This bit must be set to 1 during a write to allow modification of the ST bit. When CHG is set to 0 during a write, ST is not altered when writing the control word. This bit always reads as 0.

Bit 1: Start/Stop DMA Channel (ST)—The DMA channel is started when the start bit is set to 1. This bit can be modified only when the CHG bit is set to 1 during the same register write. This bit is 0 after processor reset.

Bit 0: Byte/Word Select (B̅/W)—On the Am186ES microcontroller, when B̅/W is set to 1, word transfers are selected. When B̅/W is set to 0, byte transfers are selected. Word transfers are not supported on the Am188ES microcontroller. Only byte transfers are supported when either the source or the destination bus width is 8 bits.

9.3.2 Serial Port/DMA Transfers

The Am186ES and Am188ES microcontrollers have the added feature of being able to DMA to and from the serial ports. This is accomplished by programming the DMA controller to perform transfers between a data buffer (located either in memory or I/O space) and a serial port peripheral control register (SP0TD, SP1TD, SP0RD, or SP1RD). It is important to note that when a DMA channel is in use by a serial port, the corresponding external DMA request signal is deactivated.

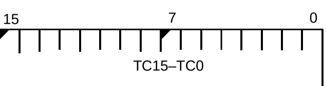
For DMA to the serial port, the transmit data register address, either I/O mapped or memory mapped, should be specified as a byte destination for the DMA by writing the address of the register into the DMA destination low and DMA destination high registers. The destination address (the address of the transmit data register) should be configured as a constant throughout the DMA operation. The serial port transmitter acts as the synchronizing device so the DMA channel should be configured as destination synchronized.

For DMA from the serial port, the receive data register address, either I/O mapped or memory mapped, should be specified as a byte source for the DMA by writing the address of the register into the DMA Source and DMA Source High registers. The source address (the address of the receive data register) should be configured as a constant throughout the DMA. The serial port receiver acts as the synchronizing device so the DMA channel should be configured as source synchronized.

9.3.3 DMA Transfer Count Registers (D0TC, Offset C8h, D1TC, Offset D8h)

Each DMA channel maintains a 16-bit DMA Transfer Count register (DTC). This register is decremented after each DMA cycle, regardless of the state of the TC bit in the DMA control register. However, if the TC bit in the DMA control word is set or if unsynchronized transfers are programmed, DMA activity terminates when the transfer count register reaches 0.

Figure 9-3 DMA Transfer Count Registers



The value of D0TC and D1TC at reset is undefined.

Bits 15–0: DMA Transfer Count (TC15–TC0)—Contains the transfer count for a DMA channel. Value is decremented by 1 after each transfer.

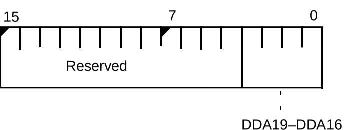


9.3.4 DMA Destination Address High Register (High Order Bits) (D0DSTH, Offset C6h, D1DSTH, Offset D6h)

Each DMA channel maintains a 20-bit destination and a 20-bit source register. Each 20-bit address takes up two full 16-bit registers (the high register and the low register) in the peripheral control block. For each DMA channel to be used, all four address registers for that channel must be initialized. These addresses can be individually incremented or decremented after each transfer. If word transfers are performed, the address is incremented or decremented by 2 after each transfer. If byte transfers are performed, the address is incremented or decremented by 1.

Each register can point into either memory or I/O space. The user must program the upper four bits to 0000b in order to address the normal 64K I/O space. Since the DMA channels can perform transfers to or from odd addresses, there is no restriction on values for the destination and source address registers. Higher transfer rates can be achieved on the Am186ES microcontroller if all word transfers are performed to or from even addresses so that accesses occur in single 16-bit bus cycles.

Figure 9-4 DMA Destination Address High Register



The value of D0DSTH and D1DSTH at reset is undefined.

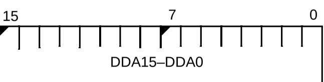
Bits 15–4: Reserved

Bits 3–0: DMA Destination Address High (DDA19–DDA16)—These bits are driven onto A19–A16 during the write phase of a DMA transfer.

9.3.5 DMA Destination Address Low Register (Low Order Bits) (D0DSTL, Offset C4h, D1DSTL, Offset D4h)

Figure 9-5 shows the DMA Destination Address Low register. The sixteen bits of this register are combined with the four bits of the DMA Destination Address High register (see Figure 9-4) to produce a 20-bit destination address.

Figure 9-5 DMA Destination Address Low Register



The value of D0DSTL and D1DSTL at reset is undefined.

Bits 15–0: DMA Destination Address Low (DDA15–DDA0)—These bits are driven onto A15–A0 during the write phase of a DMA transfer.

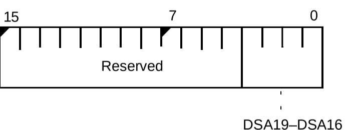


9.3.6 DMA Source Address High Register (High Order Bits) (D0SRCH, Offset C2h, D1SRCH, Offset D2h)

Each DMA channel maintains a 20-bit destination and a 20-bit source register. Each 20-bit address takes up two full 16-bit registers (the high register and the low register) in the peripheral control block. For each DMA channel to be used, all four address registers for that channel must be initialized. These addresses can be individually incremented or decremented after each transfer. If word transfers are performed, the address is incremented or decremented by 2 after each transfer. If byte transfers are performed, the address is incremented or decremented by 1.

Each register can point into either memory or I/O space. The user must program the upper four bits to 0000b in order to address the normal 64K I/O space. Since the DMA channels can perform transfers to or from odd addresses, there is no restriction on values for the destination and source address registers. Higher transfer rates can be achieved on the Am186ES microcontroller if all word transfers are performed to or from even addresses so that accesses occur in single 16-bit bus cycles.

Figure 9-6 DMA Source Address High Register



The value of D0SRCH and D1SRCH at reset is undefined.

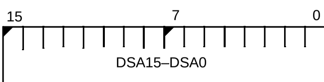
Bits 15–4: Reserved

Bits 3–0: DMA Source Address High (DSA19–DSA16)—These bits are driven onto A19–A16 during the read phase of a DMA transfer.

9.3.7 DMA Source Address Low Register (Low Order Bits) (D0SRCL, Offset C0h, D1SRCL, Offset D0h)

Figure 9-7 shows the DMA Source Address Low register. The sixteen bits of this register are combined with the four bits of the DMA Source Address High register (see Figure 9-6) to produce a 20-bit source address.

Figure 9-7 DMA Source Address Low Register



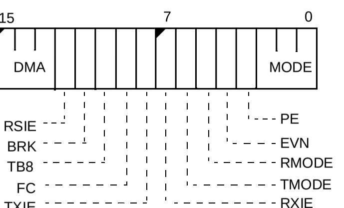
The value of D0SRCL and D1SRCL at reset is undefined.

Bits 15–0: DMA Source Address Low (DSA15–DSA0)—These bits are driven onto A15–A0 during the read phase of a DMA transfer.

10.2.1 Serial Port 0/1 Control Registers (SP0CT/SP1CT, Offset 80h/10h)

The serial port control registers control both the transmit and receive sections of the serial port. The format of the serial port control registers is shown in Figure 10-3.

Figure 10-3 Serial Port Control Register



The value of SP0CT/SP1CT at reset is 0000h.

Bits 15–13: DMA Control Field (DMA)—This field configures the serial port for use with DMA transfers according to the following table.

Table 10-3 DMA Control Bits

DMA Bits	Receive	Transmit
000b	No DMA	No DMA
001b	DMA0	DMA1
010b	DMA1	DMA0
011b	Reserved	Reserved
100b	DMA0	No DMA
101b	DMA1	No DMA
110b	No DMA	DMA0
111b	No DMA	DMA1

DMA transfers to a serial port function as destination-synchronized DMA transfers. A new transfer is requested when the transmit holding register is empty. This corresponds with the assertion of the THRE bit in the serial port status register in non-DMA mode. When the port is configured for DMA transmits, the corresponding transmit interrupt is disabled regardless of the setting of the TXIE bit.

DMA transfers from the serial port function as source-synchronized DMA transfers. A new transfer is requested when the serial port receive register contains valid data. This corresponds with the assertion of the RDR bit in the serial port status register in non-DMA mode. When the port is configured for DMA receives, the corresponding receive interrupt is disabled regardless of the setting of the RXIE bit. Receive status interrupts may still be taken, as configured by the RSIE bit.

Hardware handshaking may be used in conjunction with serial port DMA transfers.

When a DMA channel is being used for serial port transmits or receives, the DMA request is generated internally. The corresponding external DMA request signals, DRQ0 or DRQ1, are not active for serial port DMA transfers.

Bit 12: Receive Status Interrupt Enable (RSIE)—This bit enables the serial port to generate an interrupt request when an exception occurs during data reception. When this bit is set, interrupt requests are generated for the error conditions reported in the serial port status register (BRK0, BRK1, OER, PER, FER).

Bit 11: Send Break (BRK)—When this bit is set, the TXD pin is driven Low regardless of the data being shifted out of the transmit register.

A short break, as reported by the BRK0 bit in the status register, is a continuous Low on the TXD output for a duration of more than one frame transmission time M, where M = start bit + data bits (+ parity bit)+ stop bit. The transmitter can be used to time the break by setting the BRK bit when the transmitter is empty (indicated by the TEMT bit of the serial port status register), writing the serial port transmit register with data, then waiting until the TEMT bit is again set before resetting the BRK bit.

A long break, as reported by the BRK1 bit in the status register, is a continuous Low on the TXD output for a duration of more than two frame transmission times plus the transmission time for three additional bits (2M+3). The transmitter can be used to time the break as follows:

1. Wait for the TEMT bit in the status register to be set.
2. Set the BRK bit.
3. Perform two sequential writes to the transmit register.
4. Wait for the TEMT bit in the status register to be set again.
5. Write a character with the low nibble zeroed and the high nibble High (for example, F0h).
6. Clear the BRK bit. The character being transmitted continues to hold the TXD pin Low for the required additional 3-bit transmission time.

Note: *The transmitter can only be used to time the break if hardware flow control is disabled. If flow control is enabled, setting the BRK bit will still force the TXD line Low, but the receiving device may deassert the CTS input, inhibiting the clocking out of the character in the transmit data register.*

Bit 10: Transmit Bit 8 (TB8)—This bit is transmitted as the ninth data bit in modes 2 and 3 (see the mode field description). This bit is not buffered and is cleared after every transmission. In order to transmit a character with the 8th data bit High, the following protocol should be followed:

1. Wait for the TEMT bit in the status register to become set.
2. Write the control register with this bit set.
3. Write the character to be transmitted.

Bit 9: Flow Control Enable (FC)—When this bit is 1, hardware flow control is enabled for the associated serial port. When this bit is 0, hardware flow control is disabled for the associated serial port. The nature of the flow control signals is determined by the setting of the ENRX0/ENRX1 and RTS0/RTS1 bits in the AUXCON register. See the discussion of the AUXCON register and Section 10.1.1 on page 10-1 for more information. If this bit is 1 for serial port 0, the associated pins are used as flow control signals, overriding their function as Peripheral Chip Select signals. This bit is 0 after processor reset.



Bit 8: Transmitter Ready Interrupt Enable (TXIE)—When this bit is set, the serial port generates an interrupt request whenever the transmit holding register is empty (THRE bit in the status register is set), indicating that the transmitter is available to accept a new character for transmission. When this bit is reset, the serial port does not generate transmit interrupt requests. Interrupt requests continue to be generated as long as the TXIE bit is set and the transmitter does not contain valid data to transmit, i.e., the THRE bit in the status register remains set.

Bit 7: Receive Data Ready Interrupt Enable (RXIE) —When this bit is set, the serial port generates an interrupt request whenever the receive register contains valid data (RDR bit in the status register is set). When this bit is reset, the serial port does not generate receive interrupt requests. Interrupt requests continue to be generated as long the RXIE bit is set and the receiver contains unread data (the RDR bit in the status register is set).

Bit 6: Transmit Mode (TMODE)—When this bit is set, the transmit section of the serial port is enabled. When this bit is reset, the transmitter and transmit interrupt requests are disabled.

Bit 5: Receive Mode (RMODE)—When this bit is set, the receive section of the serial port is enabled. When this bit is reset, the receiver is disabled.

Bit 4: Even Parity (EVN)—This bit determines the parity sense. When EVN is set, even parity checking is enforced (even number of 1s in frame). When EVN is reset, odd parity checking is enforced (odd number of 1s in frame).

Note: *This bit is valid only when the PE bit is set (parity enabled).*

Bit 3: Parity Enable (PE)—When this bit is set, parity checking is enabled. When this bit is reset, parity checking is disabled.

Bits 2–0: Mode of Operation (MODE)—This field determines the operating mode for the serial port. The valid modes and their descriptions are shown in Table 10-4.

Mode 1 supports 7 data bits when parity is enabled or 8 data bits with parity disabled. When using parity, the eighth bit becomes the parity bit and is generated for transmits, or checked for receives automatically by the processor.

Table 10-4 Serial Port MODE Settings

MODE	Description	Data Bits	Parity Bits	Stop Bits
0	Reserved			
1	Data Mode 1	7 or 8	1 or 0	1
2	Data Mode 2	9	N/A	1
3	Data Mode 3	8 or 9	1 or 0	1
4	Data Mode 4	7	N/A	1
5	Reserved			
6	Reserved			
7	Reserved			

Mode 2—When configured in this mode, the serial port receiver will not complete a data reception unless the ninth data bit is set (High). Any character received with the ninth data

bit reset (Low) is ignored. The transmit portion of the port behaves identically with mode 3 operation.

This mode can be used in conjunction with mode 3 to allow for multidrop communications over a common serial link. In this case, the serial port is configured as mode 2 initially. Each time data is received with the ninth bit set, the data is compared by software against a unique ID for this receiver. If the received data does not match the port ID, the port is left in mode 2. If the received data matches the port ID, software should reconfigure the serial port to mode 3, allowing it to receive 9-bit data with the ninth bit reset.

In a serial multidrop configuration, multiple serial ports are attached to the same serial line. The master serial port is configured in mode 3 while the slave serial ports are configured in mode 2. The master polls the other devices by sending out status request packets. Each of these status request packets begins with an address byte (i.e. ninth data bit is set). The slave ports report a receive character for the address byte since the ninth bit is set. Each port then attempts to match the address against its own address. If the addresses do not match, the port remains in mode 2 and ignores the remainder of the message. If the addresses match, software reconfigures the port into mode 3. The two mode 3 ports are able to exchange data freely.

It should be noted that only ports which are actively exchanging data (i.e. ports in mode 3) should have hardware handshaking enabled. If this is not the case, multiple devices may be driving the hardware handshaking lines. For this reason, hardware handshaking is not supported for the mode 2 configuration and should not be enabled. In addition, if it is possible for more than two devices to be configured as mode 3 at any one time, hardware handshaking should not be enabled.

Mode 3 supports 8 data bits when parity is enabled or 9 data bits with parity disabled. When not using parity, the ninth bit (bit 8) for transmission is set by writing a 1 to the TB8 field in the serial port control register. The ninth data bit for a receive can be read in the RB8 field of the serial port status register. See the discussion of the TB8 and RB8 fields for more information.

This mode can be used in conjunction with mode 2 (see above) to allow for multidrop communications over a common serial link. In this case, parity must be disabled. In this configuration, software interprets receive characters as data as long as the ninth data bit is reset (Low). When a character is received with the ninth bit set, software should compare the lower eight bits against the port ID. If the port ID matches the receive data, the port should remain in mode 3. If the port ID does not match the receive data, the port should be reconfigured to mode 2.

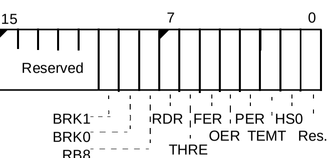
Mode 4—In this mode, each frame consists of 7 data bits, a start bit, and a stop bit. Parity is not available in this mode.



10.2.2 Serial Port 0/1 Status Registers (SP0STS/SP1STS, Offset 82h/12h)

The Serial Port Status Registers provide information about the current status of the associated serial port. The THRE and TEMT fields provide the software with information about the state of the transmitter. The BRK1, BRK0, RB8, RDR, FER, OER, and PER bits provide information about the receiver. The HS0 bit reflects the value of the serial port's associated CTS/ENRX signal. The THRE, TEMT, and HS0 bits are updated during each processor cycle. The format of the Serial Port Status Register is shown in Figure 10-4.

Figure 10-4 Serial Port 0/1 Status Register



Bits 15–11: Reserved

Bit 10: Long Break Detected (BRK1)—This bit is set when a long break is detected on the asynchronous serial interface. A long break is defined as a Low signal on the RXD pin for greater than 2M+3 bit times, where M = (start bit + # data bits + # parity bits + stop bit).

If the serial port is receiving a character when the break begins, the reception of the character will be completed (generating a framing error) before timing for the break begins. To guarantee detection with the specified 2M+3 bit times, the break must begin outside of a frame.

Note: *This bit should be reset by software.*

Bit 9: Short Break Detected (BRK0)—This bit is set when a short break is detected on the asynchronous serial interface. A short break is defined as a Low signal on the RXD pin for greater than M bit times, where M = (start bit + # data bits + # parity bits + stop bit).

If the serial port is receiving a character when the break begins, the reception of the character will be completed (generating a framing error) before timing for the break begins. To guarantee detection with the specified M bit times, the break must begin outside of a frame.

Note: *This bit should be reset by software.*

Bit 8: Received Bit 8 (RB8)—This bit contains the ninth data bit received in modes 2 and 3. (See Serial Port Control register definition.)

Note: *This bit should be reset by software.*

Bit 7: Receive Data Ready (RDR)—When this bit is set, the corresponding Receive Data register contains valid data. This field is read-only. The RDR bit can only be reset by reading the associated SP0RD/SP1RD register.

Bit 6: Transmit Holding Register Empty (THRE)—When this bit is set, the transmit holding register is ready to accept data for transmission. This field is read-only.

Bit 5: Framing Error Detected (FER)—When this bit is set, the serial port has detected a framing error. Framing errors are generated when the receiver samples the RXD line as Low when it expected the stop bit.

Note: *This bit should be reset by software.*

Bit 4: Overrun Error Detected (OER)—This bit is set when the processor detects an overrun error. An overrun error occurs when the serial port overwrites valid, unread data in the receive register, resulting in loss of data.

Note: *This bit should be reset by software.*

Bit 3: Parity Error Detected (PER)—This bit is set when the processor detects a parity error (modes 1 and 3).

Note: *This bit should be reset by software.*

Bit 2: Transmitter Empty (TEMT)—When this bit is set, the transmitter has no data to transmit and the transmit shift register is empty. This indicates to software that it is safe to disable the transmit section. This bit is read-only.

Bit 1: Handshake Signal 0 (HS0)—This bit reflects the inverted value of the external $\overline{CTS}$ pin. If $\overline{CTS}$ is asserted, HS0 is set to 1. This bit is read-only.

Bit 0: Reserved

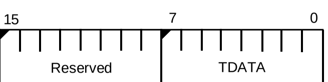
10.2.3 Serial Port 0/1 Transmit Registers (SP0TD/SP1TD, Offset 84h/14h)

The transmit registers (Figure 10-5) are written by software with the value to be transmitted over the serial interface. The transmitter is double-buffered; data to be transmitted is copied from the transmit register to the transmit shift register (which is not accessible to software) before transmitting. The state of the transmit and transmit shift registers is reflected in the TEMT and THRE bits in the associated Serial Port Status register.

When hardware handshaking is enabled, the transmitter will not transmit data while $\overline{RTS}$/$\overline{RTR}$ inputs are deasserted. Data is held in the transmit and transmit shift registers without affecting the transmit pin.

The serial port transmit register in the Am186EM and Am188EM microcontrollers is renamed in the Am186ES and Am188ES microcontrollers as the Serial Port 0 Transmit register.

Figure 10-5 Serial Port 0/1 Transmit Registers



The value of SPTD at reset is undefined.

Bits 15–8: Reserved

Bit 7–0: Transmit Data (TDATA)—This field contains data to be transmitted through the asynchronous serial port.

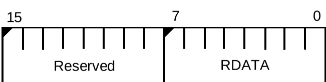
10.2.4 Serial Port 0/1 Receive Registers (SP0RD/SP1RD, Offset 86h/16h)

These registers (Figure 10-6) contain data received over the serial port. The receiver is double-buffered; the receive section can be receiving a subsequent frame of data in the receive shift register (which is not accessible to software) while the receive data register is being read.

The Receive-Data-Ready (RDR) bit in the serial port status register reports the current state of this register. When the RDR bit is set, the receive register contains valid unread data. The RDR bit is automatically cleared when the receive register is read.

When hardware handshaking is enabled, the $\overline{CTS}$/$\overline{ENRX}$ signals are deasserted while the receive register contains valid unread data. Reading the receive register causes the $\overline{CTS}$/$\overline{ENRX}$ signals to be asserted. This behavior prevents overrun errors, but may result in delays between character transmissions.

Figure 10-6 Serial Port Receive 0/1 Registers



The value of SPRD at reset is undefined.

Bits 15–8: Reserved

Bits 7–0: Receive Data (RDATA)—This field holds valid data received over the serial line only when the RDR bit in the associated serial port control register is set.

10.2.5 Serial Port 0/1 Baud Rate Divisor Registers (SP0BAUD/SP1BAUD, Offset 88h/18h)

Each of the asynchronous serial ports has a baud rate divisor register, so the two ports can operate at different rates.

These registers (Figure 10-7) specify a clock divisor for the generation of the serial clock that controls the associated serial port. The baud rate divisor register specifies the number of internal processor cycles in one phase (half period) of the 16x serial clock.

If power-save mode is in effect, the baud rate divisor must be reprogrammed to reflect the new processor clock frequency. Since power-save mode is automatically exited when an interrupt is taken, serial port transmits and receives may be corrupted if the serial port is in use and interrupts are enabled during power-save mode.

A general formula for the baud rate divisor is:

BAUDDIV =
(Processor Frequency ÷ (16 · baud rate))

The maximum baud rate is 1/16 of the internal processor clock and is achieved by setting BAUDDIV=0001h. This results in a baud rate of 2500 Kb at 20 MHz, 1562.5 Kb at 25MHz, 2062.5 Kb at 33 MHz, and 1250 Kb at 40 MHz. A BAUDDIV setting of zero results in no transmission or reception of data.

The serial port receiver can tolerate a 3.0% overspeed and 2.5% underspeed baud rate deviance.

Table 10-5 Common Baud Rates

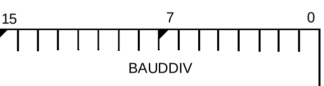
Baud Rate	Divisor Based on CPU Clock Rate			
	20 MHz	25 MHz	33 MHz	40 MHz
300	4166	5208	6875	8333
600	2083	2604	3437	4166
1050	1190	1488	1964	2380
1200	1041	1302	1718	2083
1800	694	868	1145	1388
2400	520	651	859	1041
4800	260	325	429	520
7200	173	217	286	347
9600	130	162	214	260
19200	65	81	107	130
28800	43	54	71	86
38400	33	40	53	65
56000	22	28	36	45
57600	22	27	35	43
76800	16	20	26	32
115200	10	13	18	22

Baud Rate	Divisor Based on CPU Clock Rate			
	20 MHz	25 MHz	33 MHz	40 MHz
128000	9	12	16	19
153600	8	10	13	16

Special	15 MHz	21 MHz	24 MHz	30 MHz
187500	5	7	8	10

Note: *A 1% error applies to all values in the above tables.*

Figure 10-7 Serial Port 0/1 Baud Rate Divisor Registers



The value of SPBAUD at reset is 0000h.

Bits 15–0: Baud Rate Divisor (BAUDDIV)—This field specifies the divisor for the internal processor clock.



Table 11-1 PIO Pin Assignments

PIO No	Associated Pin	Power-On Reset Status
0	TMRIN1	Input with pullup
1	TMROUT1	Input with pulldown
2	PCS6/A2	Input with pullup
3	PCS5/A1	Input with pullup
4	DT/R	Normal operation(3)
5	DEN/DS	Normal operation(3)
6	SRDY	Normal operation(4)
7(1)	A17	Normal operation(3)
8(1)	A18	Normal operation(3)
9(1)	A19	Normal operation(3)
10	TMROUT0	Input with pulldown
11	TMRIN0	Input with pullup
12	DRQ0/INT5	Input with pullup
13	DRQ1/INT6	Input with pullup
14	MCS0	Input with pullup
15	MCS1	Input with pullup
16	PCS0	Input with pullup
17	PCS1	Input with pullup
18	PCS2/CTS1/ENRX1	Input with pullup
19	PCS3/RTS1/RTR1	Input with pullup
20	RTS0/RTR0	Input with pullup
21	CTS0/ENRX0	Input with pullup
22	TXD0	Input with pullup
23	RXD0	Input with pullup
24	MCS2	Input with pullup
25	MCS3/RFSH	Input with pullup
26(1,2)	UZI	Input with pullup
27	TXD1	Input with pullup
28	RXD1	Input with pullup
29(1,2)	S6/LOCK/CLKDIV2	Input with pullup
30	INT4	Input with pullup
31	INT2/INTA0/PWD	Input with pullup

Notes:

1. *These pins are used by emulators. (Emulators also use S2–S0, RES, NMI, CLKOUTA, BHE, ALE, AD15–AD0, and A16–A0.)*
2. *These pins revert to normal operation if BHE/ADEN (186) or RFSH2/ADEN (188) is held Low during power-on reset.*
3. *When used as a PIO, input with pullup option available.*
4. *When used as a PIO, input with pulldown option available.*

11.2 PIO MODE REGISTERS

Table 11-2 shows the possible settings for the PIO Mode and PIO Direction bits. The Am186ES and Am188ES microcontrollers default the 32 PIO pins to either 00b (normal operation) or 01b (PIO input with weak internal pullup or pulldown enabled).

Pins that default to active High outputs at reset are pulled down. All other pins are pulled up or are normal operation. See Table 11-2. The column titled *Power-On Reset State* in Table 11-1 lists the defaults for the PIOs.

The internal pullup resistor has a value of approximately 10 Kohms. The internal pulldown resistor has a value of approximately 10 Kohms.

Table 11-2 PIO Mode and PIO Direction Settings

PIO Mode	PIO Direction	Pin Function
0	0	Normal operation
0	1	PIO input with pullup/pulldown
1	0	PIO output
1	1	PIO input w/o pullup/pulldown

Figure 11-2 PIO Mode 1 Register (PIOMODE1, offset 76h)

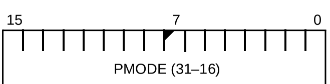
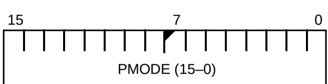


Figure 11-3 PIO Mode 0 Register (PIOMODE0, offset 70h)



11.2.1 PIO Mode 1 Register (PIOMODE1, Offset 76h)

The value of PIOMODE1 at reset is 0000h.

Bits 15–0: PIO Mode Bits (PMODE31–PMODE16)—This field, along with the PIO direction registers, determines whether each PIO pin performs its preassigned function or is enabled as a custom PIO signal. The most significant bit of the PMODE field determines whether PIO31 is enabled, the next bit determines whether PIO30 is enabled, and so on. Table 11-2 shows the values that the PIO mode bits and the PIO direction bits can encode.

11.2.2 PIO Mode 0 Register (PIOMODE0, Offset 70h)

The value of PIOMODE0 at reset is 0000h.

Bits 15–0: PIO Mode Bits (PMODE15–PMODE0)—This field is a continuation of the PMODE field in the PIO Mode 1 register.

11.3 PIO DIRECTION REGISTERS

Each PIO is individually programmed as an input or output by a bit in one of the PIO Direction registers (see Figure 11-4 and Figure 11-5). Table 11-2 on page 11-3 shows the values that the PIO mode bits and the PIO direction bits can encode. The column titled *Power-On Reset Status* in Table 11-1 lists the reset default values for the PIOs. Bits in the PIO Direction registers have the same correspondence to pins as bits in the PIO Mode registers.

Figure 11-4 PIO Direction 1 Register (PDIR1, offset 78h)

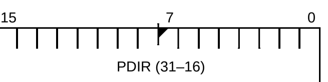


Figure 11-5 PIO Direction 0 Register (PDIR0, offset 72h)



11.3.1 PIO Direction 1 Register (PDIR1, Offset 78h)

The value of PDIR1 at reset is FFFFh.

Bits 15–0: PIO Direction Bits (PDIR31–PDIR16)—This field determines whether each PIO pin acts as an input or an output. The most significant bit of the PDIR field determines the direction of PIO31, the next bit determines the direction of PIO30, and so on. A 1 in the bit configures the PIO signal as an input and a 0 in the bit configures it as an output or as normal pin function.

11.3.2 PIO Direction 0 Register (PDIR0, Offset 72h)

The value of PDIR0 at reset is FC0Fh.

Bits 15–0: PIO Direction Bits (PDIR15–PDIR0)—This field is a continuation of the PDIR field in the PIO Direction 1 register.

11.4 PIO DATA REGISTERS

If a PIO pin is enabled as an output, the value in the corresponding bit in one of the PIO Data registers (see Figure 11-6 and Figure 11-7) is driven on the pin with no inversion (Low=0, High=1). If a PIO pin is enabled as an input, the value on the PIO pin is reflected in the value of the corresponding bit in the PIO Data register, with no inversion. Bits in the PIO Data registers have the same correspondence to pins as bits in the PIO Mode registers and PIO Direction registers.

Figure 11-6 PIO Data 1 Register (PDATA1, offset 7Ah)

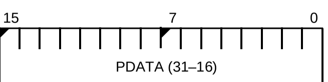
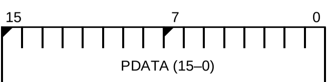


Figure 11-7 PIO Data 0 Register (PDATA0, offset 74h)



11.4.1 PIO Data Register 1 (PDATA1, Offset 7Ah)

Bits 7–0: PIO Data Bits (PDATA31–PDATA16)—This field determines the level driven on each PIO pin or reflects the external level of the pin, depending upon whether the pin is configured as an output or an input in the PIO Direction registers. The most significant bit of the PDATA field indicates the level of PIO31, the next bit indicates the level of PIO30, and so on.

The value of PDATA1 at reset is undefined.

11.4.2 PIO Data Register 0 (PDATA0, Offset 74h)

Bits 15–0: PIO Data Bits (PDATA15–PDATA0)—This field is a continuation of the PDATA field in the PIO Data 1 register.

The value of PDATA0 at reset is undefined.

11.5 OPEN-DRAIN OUTPUTS

The PIO Data registers permit the PIO signals to be operated as open-drain outputs. This is accomplished by keeping the appropriate PDATA bits constant in the PIO Data register and writing the data value into its associated bit position in the PIO Direction register, so the output is either driving Low or is disabled, depending on the data.

CONTROLLORE CAN SJA 1000

1 FEATURES

- Pin compatibility to the PCA82C200 stand-alone CAN controller
- Electrical compatibility to the PCA82C200 stand-alone CAN controller
- Software-compatibility mode to the PCA82C200 (BasicCAN mode is default)
- Extended receive buffer (64-byte FIFO)
- CAN 2.0B protocol compatibility (extended frame passive in PCA82C200 compatibility mode)
- Supports 11-bit identifier as well as 29-bit identifier
- Bit rates up to 1 Mbits/s
- PeliCAN mode extensions:
  - Error counters with read/write access
  - Programmable error warning limit
  - Last error code register
  - Error interrupt for each CAN-bus error
  - Arbitration lost interrupt with detailed bit position
  - Single-shot transmission (no re-transmission)
  - Listen only mode (no acknowledge, no active error flags)
  - Hot plugging support (software driven bit rate detection)
  - Acceptance filter extension (4-byte code, 4-byte mask)
  - Reception of ‘own’ messages (self reception request)
- 24 MHz clock frequency
- Interfaces to a variety of microprocessors
- Programmable CAN output driver configuration
- Extended ambient temperature range (–40 to +125 °C).

2 GENERAL DESCRIPTION

The SJA1000 is a stand-alone controller for the Controller Area Network (CAN) used within automotive and general industrial environments. It is designed to be hardware and software compatible to the PCA82C200 CAN controller (BasicCAN) from Philips Semiconductors. Additionally, a new mode of operation is implemented (PeliCAN) which supports the CAN 2.0B protocol specification with several new features.

3 ORDERING INFORMATION

TYPE NUMBER	PACKAGE		
	NAME	DESCRIPTION	VERSION
SJA1000	DIP28	plastic dual in-line package; 28 leads (600 mil)	SOT117-1
SJA1000T	SO28	plastic small outline package; 28 leads; body width 7.5 mm	SOT136-1

4 BLOCK DIAGRAM

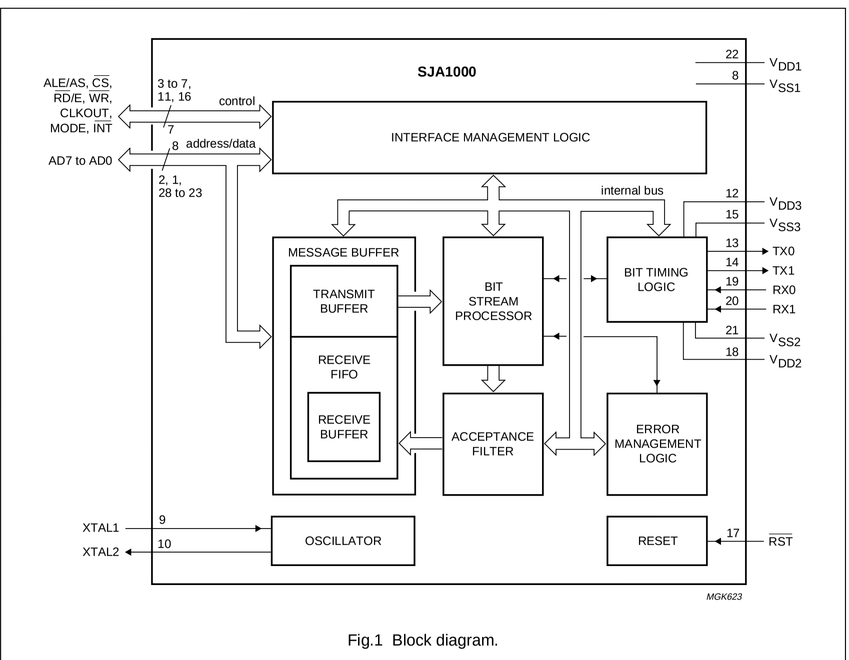


Fig.1 Block diagram.



6 FUNCTIONAL DESCRIPTION

6.1 Description of the CAN controller blocks

6.1.1 INTERFACE MANAGEMENT LOGIC (IML)

The interface management logic interprets commands from the CPU, controls addressing of the CAN registers and provides interrupts and status information to the host microcontroller.

6.1.2 TRANSMIT BUFFER (TXB)

The transmit buffer is an interface between the CPU and the Bit Stream Processor (BSP) that is able to store a complete message for transmission over the CAN network. The buffer is 13 bytes long, written to by the CPU and read out by the BSP.

6.1.3 RECEIVE BUFFER (RXB, RXFIFO)

The receive buffer is an interface between the acceptance filter and the CPU that stores the received and accepted messages from the CAN-bus line. The Receive Buffer (RXB) represents a CPU-accessible 13-byte window of the Receive FIFO (RXFIFO), which has a total length of 64 bytes.
With the help of this FIFO the CPU is able to process one message while other messages are being received.

6.1.4 ACCEPTANCE FILTER (ACF)

The acceptance filter compares the received identifier with the acceptance filter register contents and decides whether this message should be accepted or not. In the event of a positive acceptance test, the complete message is stored in the RXFIFO.

6.1.5 BIT STREAM PROCESSOR (BSP)

The bit stream processor is a sequencer which controls the data stream between the transmit buffer, RXFIFO and the CAN-bus. It also performs the error detection, arbitration, stuffing and error handling on the CAN-bus.

6.1.6 BIT TIMING LOGIC (BTL)

The bit timing logic monitors the serial CAN-bus line and handles the bus line-related bit timing. It is synchronized to the bit stream on the CAN-bus on a 'recessive-to-dominant' bus line transition at the beginning of a message (hard synchronization) and re-synchronized on further transitions during the reception of a message (soft synchronization). The BTL also provides programmable time segments to compensate for the propagation delay times and phase shifts (e.g. due to oscillator drifts) and to define the sample point and the number of samples to be taken within a bit time.

6.1.7 ERROR MANAGEMENT LOGIC (EML)

The EML is responsible for the error confinement of the transfer-layer modules. It receives error announcements from the BSP and then informs the BSP and IML about error statistics.

6.2 Detailed description of the CAN controller

The SJA1000 is designed to be software and pin-compatible to its predecessor, the PCA82C200 stand-alone CAN controller. Additionally, a lot of new functions are implemented. To achieve the software compatibility, two different modes of operation are implemented:

- BasicCAN mode; PCA82C200 compatible
- PeliCAN mode; extended features.

The mode of operation is selected with the CAN-mode bit located within the clock divider register. Default mode upon reset is the BasicCAN mode.

6.2.1 PCA82C200 COMPATIBILITY

In BasicCAN mode the SJA1000 emulates all known registers from the PCA82C200 stand-alone CAN controller. The characteristics, as described in Sections 6.2.1.1 to 6.2.1.4 are different from the PCA82C200 design with respect to software compatibility.

6.2.1.1 Synchronization mode

The SYNC bit in the control register is removed (CR.6 in the PCA82C200). Synchronization is only possible by a recessive-to-dominant transition on the CAN-bus. Writing to this bit has no effect. To achieve compatibility to existing application software, a read access to this bit will reflect the previously written value (flip-flop without effect).

6.2.1.2 Clock divider register

The clock divider register is used to select the CAN mode of operation (BasicCAN/PeliCAN). Therefore one of the reserved bits within the PCA82C200 is used. Writing a value between 0 and 7, as allowed for the PCA82C200, will enter the BasicCAN mode. The default state is divide by 12 for Motorola mode and divide by 2 for Intel mode. An additional function is implemented within another of the reserved bits. Setting of bit CBP (see Table 49) enables the internal RX input comparator to be bypassed thereby reducing the internal delays if an external transceiver circuit is used.

6.2.1.3 Receive buffer

The dual receive buffer concept of the PCA82C200 is replaced by the receive FIFO from the PeliCAN controller. This has no effect to the application software except for the data overrun probability. Now more than two messages may be received (up to 64 bytes) until a data overrun occurs.

6.2.1.4 CAN 2.0B

The SJA1000 is designed to support the full CAN 2.0B protocol specification, which means that the extended oscillator tolerance is implemented as well as the processing of extended frame messages. In BasicCAN mode it is possible to transmit and receive standard frame messages only (11-bit identifier). If extended frame messages (29-bit identifier) are detected on the CAN-bus, they are tolerated and an acknowledge is given if the message was correct, but there is no receive interrupt generated.

6.2.2 DIFFERENCES BETWEEN BASICCAN AND PELICAN MODE

In the PeliCAN mode the SJA1000 appears with a re-organized register mapping with a lot of new features. All known bits from the PCA82C200 design are available as well as several new ones. In the PeliCAN mode the complete CAN 2.0B functionality is supported (29-bit identifier).

Main new features of the SJA1000 are:

- Reception and transmission of standard and extended frame format messages
- Receive FIFO (64-byte)
- Single/dual acceptance filter with mask and code register for standard and extended frame
- Error counters with read/write access
- Programmable error warning limit
- Last error code register
- Error interrupt for each CAN-bus error
- Arbitration lost interrupt with detailed bit position
- Single-shot transmission (no re-transmission on error or arbitration lost)
- Listen only mode (monitoring of the CAN-bus, no acknowledge, no error flags)
- Hot plugging supported (disturbance-free software driven bit rate detection)
- Disable CLKOUT by hardware.

6.3 BasicCAN mode

6.3.1 BASICCAN ADDRESS LAYOUT

The SJA1000 appears to a microcontroller as a memory-mapped I/O device. An independent operation of both devices is guaranteed by a RAM-like implementation of the on-chip registers.

The address area of the SJA1000 consists of the control segment and the message buffers. The control segment is programmed during an initialization download in order to configure communication parameters (e.g. bit timing). Communication over the CAN-bus is also controlled via this segment by the microcontroller. During initialization the CLKOUT signal may be programmed to a value determined by the microcontroller.

A message, which should be transmitted, has to be written to the transmit buffer. After a successful reception the microcontroller may read the received message from the receive buffer and then release it for further use.

The exchange of status, control and command signals between the microcontroller and the SJA1000 is performed in the control segment. The layout of this segment is shown in Table 3. After an initial download, the contents of the registers acceptance code, acceptance mask, bus timing registers 0 and 1 and output control should not be changed. Therefore these registers may only be accessed when the reset request bit in the control register is set HIGH.

For register access, two different modes have to be distinguished:

- Reset mode
- Operating mode.

The reset mode (see Table 3, control register, bit Reset Request) is entered automatically after a hardware reset or when the controller enters the bus-off state (see Table 5, status register, bit Bus Status). The operating mode is activated by resetting of the reset request bit in the control register.

Table 1 BasicCAN address allocation; note 1

CAN ADDRESS	SEGMENT	OPERATING MODE		RESET MODE	
		READ	WRITE	READ	WRITE
0	control	control	control	control	control
1		(FFH)(2)	command	(FFH)(2)	command
2		status	–	status	–
3		interrupt	–	interrupt	–
4		(FFH)(2)	–	acceptance code	acceptance code
5		(FFH)(2)	–	acceptance mask	acceptance mask
6		(FFH)(2)	–	bus timing 0	bus timing 0
7		(FFH)(2)	–	bus timing 1	bus timing 1
8		(FFH)(2)	–	output control	output control
9		test	test	test	test
10	transmit buffer	identifier (10 to 3)	identifier (10 to 3)	(FFH)(2)	–
11		identifier (2 to 0), RTR and DLC	identifier (2 to 0), RTR and DLC	(FFH)(2)	–
12		data byte 1	data byte 1	(FFH)(2)	–
13		data byte 2	data byte 2	(FFH)(2)	–
14		data byte 3	data byte 3	(FFH)(2)	–
15		data byte 4	data byte 4	(FFH)(2)	–
16		data byte 5	data byte 5	(FFH)(2)	–
17		data byte 6	data byte 6	(FFH)(2)	–
18		data byte 7	data byte 7	(FFH)(2)	–
19		data byte 8	data byte 8	(FFH)(2)	–
20	receive buffer	identifier (10 to 3)	identifier (10 to 3)	identifier (10 to 3)	identifier (10 to 3)
21		identifier (2 to 0), RTR and DLC	identifier (2 to 0), RTR and DLC	identifier (2 to 0), RTR and DLC	identifier (2 to 0), RTR and DLC
22		data byte 1	data byte 1	data byte 1	data byte 1
23		data byte 2	data byte 2	data byte 2	data byte 2
24		data byte 3	data byte 3	data byte 3	data byte 3
25		data byte 4	data byte 4	data byte 4	data byte 4
26		data byte 5	data byte 5	data byte 5	data byte 5
27		data byte 6	data byte 6	data byte 6	data byte 6
28		data byte 7	data byte 7	data byte 7	data byte 7
29		data byte 8	data byte 8	data byte 8	data byte 8
30		(FFH)(2)	–	(FFH)(2)	–
31		clock divider	clock divider; note 3	clock divider	clock divider

Notes

1. It should be noted that the registers are repeated within higher CAN address areas (the most significant bits of the 8-bit CPU address are not decoded: CAN address 32 continues with CAN address 0 and so on).
2. During read-out of this register a zero is always given.
3. Some bits are writeable in reset mode only (CAN mode and CBP).

Philips Semiconductors Preliminary specification

Stand-alone CAN controller SJA1000

6.3.2 RESET VALUES

Detection of a 'reset request' results in aborting the current transmission/reception of a message and entering the reset mode. On the '1-to-0' transition of the reset request bit, the CAN controller returns to the operating mode.

Table 2 Reset mode configuration; notes 1 and 2

REGISTER	BIT	SYMBOL	NAME	VALUE	
				RESET BY HARDWARE	SETTING BIT CR.0 BY SOFTWARE OR DUE TO BUS-OFF
Control	CR.7	–	reserved	0	0
	CR.6	–	reserved	X	X
	CR.5	–	reserved	1	1
	CR.4	OIE	Overrun Interrupt Enable	X	X
	CR.3	EIE	Error Interrupt Enable	X	X
	CR.2	TIE	Transmit Interrupt Enable	X	X
	CR.1	RIE	Receive Interrupt Enable	X	X
	CR.0	RR	Reset Request	1 (reset mode)	1 (reset mode)
Command	CMR.7	–	reserved	note 3	note 3
	CMR.6	–	reserved		
	CMR.5	–	reserved		
	CMR.4	GTS	Go To Sleep		
	CMR.3	CDO	Clear Data Overrun		
	CMR.2	RRB	Release Receive Buffer		
	CMR.1	AT	Abort Transmission		
	CMR.0	TR	Transmission Request		
Status	SR.7	BS	Bus Status	0 (bus-on)	X
	SR.6	ES	Error Status	0 (ok)	X
	SR.5	TS	Transmit Status	0 (idle)	0 (idle)
	SR.4	RS	Receive Status	0 (idle)	0 (idle)
	SR.3	TCS	Transmission Complete Status	1 (complete)	X
	SR.2	TBS	Transmit Buffer Status	1 (released)	1 (released)
	SR.1	DOS	Data Overrun Status	0 (absent)	0 (absent)
	SR.0	RBS	Receive Buffer Status	0 (empty)	0 (empty)
Interrupt	IR.7	–	reserved	1	1
	IR.6	–	reserved	1	1
	IR.5	–	reserved	1	1
	IR.4	WUI	Wake-Up Interrupt	0 (reset)	0 (reset)
	IR.3	DOI	Data Overrun Interrupt	0 (reset)	0 (reset)
	IR.2	EI	Error Interrupt	0 (reset)	X; note 4
	IR.1	TI	Transmit Interrupt	0 (reset)	0 (reset)
	IR.0	RI	Receive Interrupt	0 (reset)	0 (reset)

Stand-alone CAN controller — SJA1000

REGISTER	BIT	SYMBOL	NAME	VALUE	
				RESET BY HARDWARE	SETTING BIT CR.0 BY SOFTWARE OR DUE TO BUS-OFF
Acceptance code	AC.7 to 0	AC	Acceptance Code	X	X
Acceptance mask	AM.7 to 0	AM	Acceptance Mask	X	X
Bus timing 0	BTR0.7	SJW.1	Synchronization Jump Width 1	X	X
	BTR0.6	SJW.0	Synchronization Jump Width 0	X	X
	BTR0.5	BRP.5	Baud Rate Prescaler 5	X	X
	BTR0.4	BRP.4	Baud Rate Prescaler 4	X	X
	BTR0.3	BRP.3	Baud Rate Prescaler 3	X	X
	BTR0.2	BRP.2	Baud Rate Prescaler 2	X	X
	BTR0.1	BRP.1	Baud Rate Prescaler 1	X	X
	BTR0.0	BRP.0	Baud Rate Prescaler 0	X	X
Bus timing 1	BTR1.7	SAM	Sampling	X	X
	BTR1.6	TSEG2.2	Time Segment 2.2	X	X
	BTR1.5	TSEG2.1	Time Segment 2.1	X	X
	BTR1.4	TSEG2.0	Time Segment 2.0	X	X
	BTR1.3	TSEG1.3	Time Segment 1.3	X	X
	BTR1.2	TSEG1.2	Time Segment 1.2	X	X
	BTR1.1	TSEG1.1	Time Segment 1.1	X	X
	BTR1.0	TSEG1.0	Time Segment 1.0	X	X
Output control	OC.7	OCTP1	Output Control Transistor P1	X	X
	OC.6	OCTN1	Output Control Transistor N1	X	X
	OC.5	OCPOL1	Output Control Polarity 1	X	X
	OC.4	OCTP0	Output Control Transistor P0	X	X
	OC.3	OCTN0	Output Control Transistor N0	X	X
	OC.2	OCPOL0	Output Control Polarity 0	X	X
	OC.1	OCMODE1	Output Control Mode 1	X	X
	OC.0	OCMODE0	Output Control Mode 0	X	X
Transmit buffer	–	TXB	Transmit Buffer	X	X
Receive buffer	–	RXB	Receive Buffer	X; note 5	X; note 5
Clock divider	–	CDR	Clock Divider Register	00000000 (Intel); 00000101 (Motorola)	X

Stand-alone CAN controller — SJA1000

Notes

1. X means that the value of these registers or bits is not influenced.
2. Remarks in brackets explain functional meaning.
3. Reading the command register will always reflect a binary '11111111'.
4. On bus-off the error interrupt is set, if enabled.
5. Internal read/write pointers of the RXFIFO are reset to their initial values. A subsequent read access to the RXB would show undefined data values (parts of old messages). If a message is transmitted, this message is written in parallel to the receive buffer but no receive interrupt is generated and the receive buffer area is not locked. So, even if the receive buffer is empty, the last transmitted message may be read from the receive buffer until it is overridden by the next received or transmitted message.
   Upon a hardware reset, the RXFIFO pointers are reset to the physical RAM address '0'. Setting CR.0 by software or due to the bus-off event will reset the RXFIFO pointers to the currently valid FIFO start address which is different from the RAM address '0' after the first release receive buffer command.

6.3.3 CONTROL REGISTER (CR)

The contents of the control register are used to change the behaviour of the CAN controller. Bits may be set or reset by the attached microcontroller which uses the control register as a read/write memory.

Table 3 Bit interpretation of the control register (CR); CAN address 0

BIT	SYMBOL	NAME	VALUE	FUNCTION
CR.7	–	–	–	reserved; note 1
CR.6	–	–	–	reserved; note 2
CR.5	–	–	–	reserved; note 3
CR.4	OIE	Overrun Interrupt Enable	1	enabled; if the data overrun bit is set, the microcontroller receives an overrun interrupt signal (see also status register; Table 5)
			0	disabled; the microcontroller receives no overrun interrupt signal from the SJA1000
CR.3	EIE	Error Interrupt Enable	1	enabled; if the error or bus status change, the microcontroller receives an error interrupt signal (see also status register; Table 5)
			0	disabled; the microcontroller receives no error interrupt signal from the SJA1000
CR.2	TIE	Transmit Interrupt Enable	1	enabled; when a message has been successfully transmitted or the transmit buffer is accessible again, (e.g. after an abort transmission command) the SJA1000 transmits a transmit interrupt signal to the microcontroller
			0	disabled; the microcontroller receives no transmit interrupt signal from the SJA1000
CR.1	RIE	Receive Interrupt Enable	1	enabled; when a message has been received without errors, the SJA1000 transmits a receive interrupt signal to the microcontroller
			0	disabled; the microcontroller receives no transmit interrupt signal from the SJA1000



BIT	SYMBOL	NAME	VALUE	FUNCTION
CR.0	RR	Reset Request; note 4	1	present; detection of a reset request results in aborting the current transmission/reception of a message and entering the reset mode
			0	absent; on the '1-to-0' transition of the reset request bit, the SJA1000 returns to the operating mode

Notes

1. Any write access to the control register has to set this bit to logic 0 (reset value is logic 0).
2. In the PCA82C200 this bit was used to select the synchronization mode. Because this mode is not longer implemented, setting this bit has no influence on the microcontroller. Due to software compatibility setting this bit is allowed. This bit will not change after hardware or software reset. In addition the value written by users software is reflected.
3. Reading this bit will always reflect a logic 1.
4. During a hardware reset or when the bus status bit is set to logic 1 (bus-off), the reset request bit is set to logic 1 (present). If this bit is accessed by software, a value change will become visible and takes effect first with the next positive edge of the internal clock which operates with $\frac{1}{2}$ of the external oscillator frequency. During an external reset the microcontroller cannot set the reset request bit to logic 0 (absent). Therefore, after having set the reset request bit to logic 0, the microcontroller must check this bit to ensure that the external reset pin is not being held HIGH. Changes of the reset request bit are synchronized with the internal divided clock. Reading the reset request bit reflects the synchronized status.
   After the reset request bit is set to logic 0 the SJA1000 will wait for:
   a) One occurrence of bus-free signal (11 recessive bits), if the preceding reset request has been caused by a hardware reset or a CPU-initiated reset
   b) 128 occurrences of bus-free, if the preceding reset request has been caused by a CAN controller initiated bus-off, before re-entering the bus-on mode; it should be noted that several registers are modified if the reset request bit was set (see also Table 2).

6.3.4 COMMAND REGISTER (CMR)

A command bit initiates an action within the transfer layer of the SJA1000. The command register appears to the microcontroller as a write only memory. If a read access is performed to this address the byte '11111111' is returned. Between two commands at least one internal clock cycle is needed to process. The internal clock is divided by two from the external oscillator frequency.

Table 4 Bit interpretation of the command register (CMR); CAN address 1

BIT	SYMBOL	NAME	VALUE	FUNCTION
CMR.7	–	–	–	reserved
CMR.6	–	–	–	reserved
CMR.5	–	–	–	reserved
CMR.4	GTS	Go To Sleep; note 1	1	sleep; the SJA1000 enters sleep mode if no CAN interrupt is pending and there is no bus activity
			0	wake up; SJA1000 operates normal
CMR.3	CDO	Clear Data Overrun; note 2	1	clear; data overrun status bit is cleared
			0	no action
CMR.2	RRB	Release Receive Buffer; note 3	1	released; the receive buffer, representing the message memory space in the RXFIFO is released
			0	no action
CMR.1	AT	Abort Transmission; note 4	1	present; if not already in progress, a pending transmission request is cancelled
			0	absent; no action
CMR.0	TR	Transmission Request; note 5	1	present; a message will be transmitted
			0	absent; no action

Notes

1. The SJA1000 will enter sleep mode if the sleep bit is set to logic 1 (sleep); there is no bus activity and no interrupt is pending. Setting of GTS with at least one of the previously mentioned exceptions valid will result in a wake-up interrupt. After sleep mode is set, the CLKOUT signal continues until at least 15 bit times have passed, to allow a host microcontroller clocked via this signal to enter its own standby mode before the CLKOUT goes LOW.
   The SJA1000 will wake up when one of the three previously mentioned conditions is negated: after 'Go To Sleep' is set LOW (wake-up), there is bus activity or $\overline{INT}$ is driven LOW (active). On wake-up, the oscillator is started and a wake-up interrupt is generated. A sleeping SJA1000 which wakes up due to bus activity will not be able to receive this message until it detects 11 consecutive recessive bits (bus-free sequence). It should be noted that setting of GTS is not possible in reset mode. After clearing of reset request, setting of GTS is possible first, when bus-free is detected again.
2. This command bit is used to clear the data overrun condition indicated by the data overrun status bit. As long as the data overrun status bit is set no further data overrun interrupt is generated. It is allowed to give the clear data overrun command at the same time as a release receive buffer command.
3. After reading the contents of the receive buffer, the microcontroller can release this memory space of the RXFIFO by setting the release receive buffer bit to logic 1. This may result in another message becoming immediately available within the receive buffer. This event will force another receive interrupt, if enabled. If there is no other message available no further receive interrupt is generated and the receive buffer status bit is cleared.
4. The abort transmission bit is used when the CPU requires the suspension of the previously requested transmission, e.g. to transmit a more urgent message before. A transmission already in progress is not stopped. In order to see if the original message had been either transmitted successfully or aborted, the transmission complete status bit should be checked. This should be done after the transmit buffer status bit has been set to logic 1 (released) or a transmit interrupt has been generated.
5. If the transmission request was set to logic 1 in a previous command, it cannot be cancelled by setting the transmission request bit to logic 0. The requested transmission may be cancelled by setting the abort transmission bit to logic 1.



6.3.5 STATUS REGISTER (SR)

The content of the status register reflects the status of the SJA1000. The status register appears to the microcontroller as a read only memory.

Table 5 Bit interpretation of the status register (SR); CAN address 2

BIT	SYMBOL	NAME	VALUE	FUNCTION
SR.7	BS	Bus Status; note 1	1	bus-off; the SJA1000 is not involved in bus activities
			0	bus-on; the SJA1000 is involved in bus activities
SR.6	ES	Error Status; note 2	1	error; at least one of the error counters has reached or exceeded the CPU warning limit
			0	ok; both error counters are below the warning limit
SR.5	TS	Transmit Status; note 3	1	transmit; the SJA1000 is transmitting a message
			0	idle; no transmit message is in progress
SR.4	RS	Receive Status; note 3	1	receive; the SJA1000 is receiving a message
			0	idle; no receive message is in progress
SR.3	TCS	Transmission Complete Status; note 4	1	complete; the last requested transmission has been successfully completed
			0	incomplete; the previously requested transmission is not yet completed
SR.2	TBS	Transmit Buffer Status; note 5	1	released; the CPU may write a message into the transmit buffer
			0	locked; the CPU cannot access the transmit buffer; a message is waiting for transmission or is already in process
SR.1	DOS	Data Overrun Status; note 6	1	overrun; a message was lost because there was not enough space for that message in the RXFIFO
			0	absent; no data overrun has occurred since the last clear data overrun command was given
SR.0	RBS	Receive Buffer Status; note 7	1	full; one or more messages are available in the RXFIFO
			0	empty; no message is available

Notes

1. When the transmit error counter exceeds the limit of 255 [the bus status bit is set to logic 1 (bus-off)] the CAN controller will set the reset request bit to logic 1 (present) and an error interrupt is generated, if enabled. It will stay in this mode until the CPU clears the reset request bit. Once this is completed the CAN controller will wait the minimum protocol-defined time (128 occurrences of the bus-free signal). After that the bus status bit is cleared (bus-on), the error status bit is set to logic 0 (ok), the error counters are reset and an error interrupt is generated, if enabled.
2. Errors detected during reception or transmission will affect the error counters according to the CAN 2.0B protocol specification. The error status bit is set when at least one of the error counters has reached or exceeded the CPU warning limit of 96. An error interrupt is generated, if enabled.
3. If both the receive status and the transmit status bits are logic 0 (idle) the CAN-bus is idle.
4. The transmission complete status bit is set to logic 0 (incomplete) whenever the transmission request bit is set to logic 1. The transmission complete status bit will remain at logic 0 (incomplete) until a message is transmitted successfully.
5. If the CPU tries to write to the transmit buffer when the transmit buffer status bit is at logic 0 (locked), the written byte will not be accepted and will be lost without being indicated.
6. When a message that shall be received has passed the acceptance filter successfully (i.e. earliest after arbitration field), the CAN controller needs space in the RXFIFO to store the message descriptor. Accordingly there must be enough space for each data byte which has been received. If there is not enough space to store the message, that message will be dropped and the data overrun condition will be indicated to the CPU only, if this received message has no errors until the last but one bit of end of frame (message becomes valid).
7. After reading a message stored in the RXFIFO and releasing this memory space with the command release receive buffer, this bit is cleared. If there is another message available within the FIFO this bit is set again with the next bit quantum (t_{scl}).

6.3.6 INTERRUPT REGISTER (IR)

The interrupt register allows the identification of an interrupt source. When one or more bits of this register are set, the INT pin is activated (LOW). After this register is read by the microcontroller, all bits are reset what results in a floating level at INT. The interrupt register appears to the microcontroller as a read only memory.

Table 6 Bit interpretation of the interrupt register (IR); CAN address 3

BIT	SYMBOL	NAME	VALUE	FUNCTION
IR.7	–	–	–	reserved
IR.6	–	–	–	reserved
IR.5	–	–	–	reserved
IR.4	WUI	Wake-Up Interrupt; note 1	1	set; this bit is set when the sleep mode is left
			0	reset; this bit is cleared by any read access of the microcontroller
IR.3	DOI	Data Overrun Interrupt; note 2	1	set; this bit is set on a '0-to-1' transition of the data overrun status bit, when the data overrun interrupt enable is set to logic 1 (enabled)
			0	reset; this bit is cleared by any read access of the microcontroller
IR.2	EI	Error Interrupt	1	set; this bit is set on a change of either the error status or bus status bits if the error interrupt enable is set to logic 1 (enabled)
			0	reset; this bit is cleared by any read access of the microcontroller
IR.1	TI	Transmit Interrupt	1	set; this bit is set whenever the transmit buffer status changes from logic 0 to logic 1 (released) and transmit interrupt enable is set to logic 1 (enabled)
			0	reset; this bit is cleared by any read access of the microcontroller
IR.0	RI	Receive Interrupt; note 3	1	set; this bit is set while the receive FIFO is not empty and the receive interrupt enable bit is set to logic 1 (enabled)
			0	reset; this bit is cleared by any read access of the microcontroller

Notes

1. A wake-up interrupt is also generated if the CPU tries to set go to sleep while the CAN controller is involved in bus activities or a CAN interrupt is pending.
2. The overrun interrupt bit (if enabled) and the data overrun status bit are set at the same time.
3. The receive interrupt bit (if enabled) and the receive buffer status bit are set at the same time.
It should be noted that the receive interrupt bit is cleared upon a read access, even if there is another message available within the FIFO. The moment the release receive buffer command is given and there is another message valid within the receive buffer, the receive interrupt is set again (if enabled) with the next t_{scl}.

6.3.7 TRANSMIT BUFFER LAYOUT

The global layout of the transmit buffer is shown in Table 7. The buffer serves to store a message from the microcontroller to be transmitted by the SJA1000. It is subdivided into a descriptor and data field. The transmit buffer can be written to and read out by the microcontroller in operating mode only. In reset mode a 'FFH' is reflected for all bytes.

Table 7 Layout of transmit buffer

CAN ADDRESS	FIELD	NAME	BITS							
			7	6	5	4	3	2	1	0
10	descriptor	identifier byte 1	ID.10	ID.9	ID.8	ID.7	ID.6	ID.5	ID.4	ID.3
11		identifier byte 2	ID.2	ID.1	ID.0	RTR	DLC.3	DLC.2	DLC.1	DLC.0
12	data	TX data 1	transmit data byte 1							
13		TX data 2	transmit data byte 2							
14		TX data 3	transmit data byte 3							
15		TX data 4	transmit data byte 4							
16		TX data 5	transmit data byte 5							
17		TX data 6	transmit data byte 6							
18		TX data 7	transmit data byte 7							
19		TX data 8	transmit data byte 8							

6.3.7.1 Identifier (ID)

The identifier consists of 11 bits (ID.10 to ID.0). ID.10 is the most significant bit, which is transmitted first on the bus during the arbitration process. The identifier acts as the message's name. It is used in a receiver for acceptance filtering and also determining the bus access priority during the arbitration process. The lower the binary value of the identifier the higher the priority. This is due to a larger number of leading dominant bits during arbitration.

6.3.7.2 Remote Transmission Request (RTR)

If this bit is set, a remote frame will be transmitted via the bus. This means that no data bytes are included within this frame. Nevertheless, it is necessary to specify the correct data length code which depends on the corresponding data frame with the same identifier coding.

If the RTR bit is not set, a data frame will be sent including the number of data bytes as specified by the data length code.

6.3.7.3 Data Length Code (DLC)

The number of bytes in the data field of a message is coded by the data length code. At the start of a remote frame transmission the data length code is not considered due to the RTR bit being at logic 1 (remote). This forces the number of transmitted/received data bytes to be logic 0. Nevertheless, the data length code must be specified correctly to avoid bus errors if two CAN controllers start a remote frame transmission with the same identifier simultaneously.

The range of the data byte count is 0 to 8 bytes and is coded as follows:

DataByteCount = $8 \times$ DLC.3 + $4 \times$ DLC.2 + $2 \times$ DLC.1 + DLC.0

For reasons of compatibility no data length code >8 should be used. If a value >8 is selected, 8 bytes are transmitted in the data frame with the data length code specified in DLC.

6.3.7.4 Data field

The number of transferred data bytes is determined by the data length code. The first bit transmitted is the most significant bit of data byte 1 at address 12.

6.3.8 RECEIVE BUFFER

The global layout of the receive buffer is very similar to the transmit buffer described in Section 6.3.7. The receive buffer is the accessible part of the RXFIFO and is located in the range between CAN address 20 and 29.



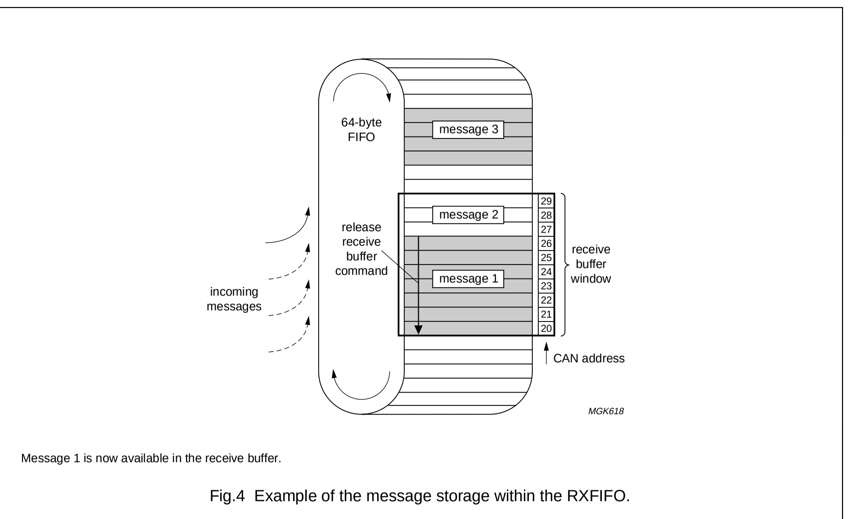


Fig.4 Example of the message storage within the RXFIFO.

Identifier, remote transmission request bit and data length code have the same meaning and location as described in the transmit buffer but within the address range 20 to 29.

As illustrated in Fig.4 the RXFIFO has space for 64 message bytes in total. The number of messages that can be stored in the FIFO at any particular moment depends on the length of the individual messages. If there is not enough space for a new message within the RXFIFO, the CAN controller generates a data overrun condition. A message which is partly written into the RXFIFO, when the data overrun condition occurs, is deleted. This situation is indicated to the microcontroller via the status register and the data overrun interrupt, if enabled and the frame was received without any errors until the last but one bit of end of frame (RX message becomes valid).

6.3.9 ACCEPTANCE FILTER

With the help of the acceptance filter the CAN controller is able to allow passing of received messages to the RXFIFO only when the identifier bits of the received message are equal to the predefined ones within the acceptance filter registers. The acceptance filter is defined by the acceptance code register (ACR; see Section 6.3.9.1) and the acceptance mask register (AMR; see Section 6.3.9.2).

6.3.9.1 Acceptance Code Register (ACR)

Table 8 ACR bit allocation; can address 4

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
AC.7	AC.6	AC.5	AC.4	AC.3	AC.2	AC.1	AC.0

This register can be accessed (read/write), if the reset request bit is set HIGH (present). When a message is received which passes the acceptance test and there is receive buffer space left, then the respective descriptor and data field are sequentially stored in the RXFIFO. When the complete message has been correctly received the following occurs:

- The receive status bit is set HIGH (full)
- If the receive interrupt enable bit is set HIGH (enabled), the receive interrupt is set HIGH (set).

6.3.9.2 Acceptance Mask Register (AMR)

Table 9 AMR bit allocation; CAN address 5

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
AM.7	AM.6	AM.5	AM.4	AM.3	AM.2	AM.1	AM.0

This register can be accessed (read/write), if the reset request bit is set HIGH (present). The acceptance mask register qualifies which of the corresponding bits of the acceptance code are 'relevant' (AM.X = 0) or 'don't care' (AM.X = 1) for acceptance filtering.

The acceptance code bits (AC.7 to AC.0) and the eight most significant bits of the message's identifier (ID.10 to ID.3) must be equal to those bit positions which are marked relevant by the acceptance mask bits (AM.7 to AM.0). If the conditions as described in the following equation are fulfilled, acceptance is given:

(ID.10 to ID.3) ≡ (AC.7 to AC.0)] ∨ (AM.7 to AM.0) ≡ 11111111

6.3.9.3 Other registers

The other registers are described in Section 6.5.

6.4 PeliCAN mode

6.4.1 PELICAN ADDRESS LAYOUT

The CAN controller's internal registers appear to the CPU as on-chip memory mapped peripheral registers. Because the CAN controller can operate in different modes (operating/reset; see also Section 6.4.3), one has to distinguish between different internal address definitions.

Starting from CAN address 32 the complete internal RAM (80-byte) is mapped to the CPU interface.



Table 10 PeliCAN address allocation; note 1

CAN ADDRESS	OPERATING MODE				RESET MODE	
	READ		WRITE		READ	WRITE
0	mode		mode		mode	mode
1	(00H)		command		(00H)	command
2	status		–		status	–
3	interrupt		–		interrupt	–
4	interrupt enable		interrupt enable		interrupt enable	interrupt enable
5	reserved (00H)		–		reserved (00H)	–
6	bus timing 0		–		bus timing 0	bus timing 0
7	bus timing 1		–		bus timing 1	bus timing 1
8	output control		–		output control	output control
9	test		test		test	test
10	reserved (00H)		–		reserved (00H)	–
11	arbitration lost capture		–		arbitration lost capture	–
12	error code capture		–		error code capture	–
13	error warning limit		–		error warning limit	error warning limit
14	RX error counter		–		RX error counter	RX error counter
15	TX error counter		–		TX error counter	TX error counter
16	RX frame information SFF; note 2	RX frame information EFF; note 3	TX frame information SFF; note 2	TX frame information EFF; note 3	acceptance code 0	acceptance code 0
17	RX identifier 1	RX identifier 1	TX identifier 1	TX identifier 1	acceptance code 1	acceptance code 1
18	RX identifier 2	RX identifier 2	TX identifier 2	TX identifier 2	acceptance code 2	acceptance code 2
19	RX data 1	RX identifier 3	TX data 1	TX identifier 3	acceptance code 3	acceptance code 3
20	RX data 2	RX identifier 4	TX data 2	TX identifier 4	acceptance mask 0	acceptance mask 0
21	RX data 3	RX data 1	TX data 3	TX data 1	acceptance mask 1	acceptance mask 1
22	RX data 4	RX data 2	TX data 4	TX data 2	acceptance mask 2	acceptance mask 2
23	RX data 5	RX data 3	TX data 5	TX data 3	acceptance mask 3	acceptance mask 3
24	RX data 6	RX data 4	TX data 6	TX data 4	reserved (00H)	–
25	RX data 7	RX data 5	TX data 7	TX data 5	reserved (00H)	–
26	RX data 8	RX data 6	TX data 8	TX data 6	reserved (00H)	–
27	(FIFO RAM); note 4	RX data 7	–	TX data 7	reserved (00H)	–
28	(FIFO RAM); note 4	RX data 8	–	TX data 8	reserved (00H)	–
29	RX message counter		–		RX message counter	–
30	RX buffer start address		–		RX buffer start address	RX buffer start address
31	clock divider		clock divider; note 5		clock divider	clock divider
32	internal RAM address 0 (FIFO)		–		internal RAM address 0	internal RAM address 0
33	internal RAM address 1 (FIFO)		–		internal RAM address 1	internal RAM address 1
↓	↓		↓		↓	↓
95	internal RAM address 63 (FIFO)		–		internal RAM address 63	internal RAM address 63
96	internal RAM address 64 (TX buffer)		–		internal RAM address 64	internal RAM address 64
↓	↓		↓		↓	↓
108	internal RAM address 76 (TX buffer)		–		internal RAM address 76	internal RAM address 76
109	internal RAM address 77 (free)		–		internal RAM address 77	internal RAM address 77
110	internal RAM address 78 (free)		–		internal RAM address 78	internal RAM address 78
111	internal RAM address 79 (free)		–		internal RAM address 79	internal RAM address 79
112	(00H)		–		(00H)	–
↓	↓		↓		↓	↓
127	(00H)		–		(00H)	–

Notes

1. It should be noted that the registers are repeated within higher CAN address areas (the most significant bit of the 8-bit CPU address is not decoded: CAN address 128 continues with CAN address 0 and so on).
2. SFF = Standard Frame Format.
3. EFF = Extended Frame Format.
4. These address allocations reflect the FIFO RAM space behind the current message. The contents are random after power-up and contain the beginning of the next message which is received after the current one. If no further message is received, parts of old messages may occur here.
5. Some bits are writeable in reset mode only (CAN mode, CBP, RXINTEN and clock off).



6.4.2 RESET VALUES

Detection of a set reset mode bit results in aborting the current transmission/reception of a message and entering the reset mode. On the '1-to-0' transition of the reset mode bit, the CAN controller returns to the mode defined within the mode register.

Table 11 Reset mode configuration; notes 1 and 2

REGISTER	BIT	SYMBOL	NAME	VALUE	
				RESET BY HARDWARE	SETTING MOD.0 BY SOFTWARE OR DUE TO BUS-OFF
Mode	MOD.7 to 5	–	reserved	0 (reserved)	0 (reserved)
	MOD.4	SM	Sleep Mode	0 (wake-up)	0 (wake-up)
	MOD.3	AFM	Acceptance Filter Mode	0 (dual)	X
	MOD.2	STM	Self Test Mode	0 (normal)	X
	MOD.1	LOM	Listen Only Mode	0 (normal)	X
	MOD.0	RM	Reset Mode	1 (present)	1 (present)
Command	CMR.7 to 5	–	reserved	0 (reserved)	0 (reserved)
	CMR.4	SRR	Self Reception Request	0 (absent)	0 (absent)
	CMR.3	CDO	Clear Data Overrun	0 (no action)	0 (no action)
	CMR.2	RRB	Release Receive Buffer	0 (no action)	0 (no action)
	CMR.1	AT	Abort Transmission	0 (absent)	0 (absent)
	CMR.0	TR	Transmission Request	0 (absent)	0 (absent)
Status	SR.7	BS	Bus Status	0 (bus-on)	X
	SR.6	ES	Error Status	0 (ok)	X
	SR.5	TS	Transmit Status	1 (wait idle)	1 (wait idle)
	SR.4	RS	Receive Status	1 (wait idle)	1 (wait idle)
	SR.3	TCS	Transmission Complete Status	1 (complete)	X
	SR.2	TBS	Transmit Buffer Status	1 (released)	1 (released)
	SR.1	DOS	Data Overrun Status	0 (absent)	0 (absent)
	SR.0	RBS	Receive Buffer Status	0 (empty)	0 (empty)
Interrupt	IR.7	BEI	Bus Error Interrupt	0 (reset)	0 (reset)
	IR.6	ALI	Arbitration Lost Interrupt	0 (reset)	0 (reset)
	IR.5	EPI	Error Passive Interrupt	0 (reset)	0 (reset)
	IR.4	WUI	Wake-Up Interrupt	0 (reset)	0 (reset)
	IR.3	DOI	Data Overrun Interrupt	0 (reset)	0 (reset)
	IR.2	EI	Error Warning Interrupt	0 (reset)	X; note 3
	IR.1	TI	Transmit Interrupt	0 (reset)	0 (reset)
	IR.0	RI	Receive Interrupt	0 (reset)	0 (reset)

REGISTER	BIT	SYMBOL	NAME	VALUE	
				RESET BY HARDWARE	SETTING MOD.0 BY SOFTWARE OR DUE TO BUS-OFF
Interrupt enable	IER.7	BEIE	Bus Error Interrupt Enable	X	X
	IER.6	ALIE	Arbitration Lost Interrupt Enable	X	X
	IER.5	EPIE	Error Passive Interrupt Enable	X	X
	IER.4	WUIE	Wake-Up Interrupt Enable	X	X
	IER.3	DOIE	Data Overrun Interrupt Enable	X	X
	IER.2	EIE	Error Warning Interrupt Enable	X	X
	IER.1	TIE	Transmit Interrupt Enable	X	X
	IER.0	RIE	Receive Interrupt Enable	X	X
Bus timing 0	BTR0.7	SJW.1	Synchronization Jump Width 1	X	X
	BTR0.6	SJW.0	Synchronization Jump Width 0	X	X
	BTR0.5	BRP.5	Baud Rate Prescaler 5	X	X
	BTR0.4	BRP.4	Baud Rate Prescaler 4	X	X
	BTR0.3	BRP.3	Baud Rate Prescaler 3	X	X
	BTR0.2	BRP.2	Baud Rate Prescaler 2	X	X
	BTR0.1	BRP.1	Baud Rate Prescaler 1	X	X
	BTR0.0	BRP.0	Baud Rate Prescaler 0	X	X
Bus timing 1	BTR1.7	SAM	Sampling	X	X
	BTR1.6	TSEG2.2	Time Segment 2.2	X	X
	BTR1.5	TSEG2.1	Time Segment 2.1	X	X
	BTR1.4	TSEG2.0	Time Segment 2.0	X	X
	BTR1.3	TSEG1.3	Time Segment 1.3	X	X
	BTR1.2	TSEG1.2	Time Segment 1.2	X	X
	BTR1.1	TSEG1.1	Time Segment 1.1	X	X
	BTR1.0	TSEG1.0	Time Segment 1.0	X	X

REGISTER	BIT	SYMBOL	NAME	VALUE	
				RESET BY HARDWARE	SETTING MOD.0 BY SOFTWARE OR DUE TO BUS-OFF
Output control	OCR.7	OCTP1	Output Control Transistor P1	X	X
	OCR.6	OCTN1	Output Control Transistor N1	X	X
	OCR.5	OCPOL1	Output Control Polarity 1	X	X
	OCR.4	OCTP0	Output Control Transistor P0	X	X
	OCR.3	OCTN0	Output Control Transistor N0	X	X
	OCR.2	OCPOL0	Output Control Polarity 0	X	X
	OCR.1	OCMODE1	Output Control Mode 1	X	X
	OCR.0	OCMODE0	Output Control Mode 0	X	X
Arbitration lost capture	–	ALC	Arbitration Lost Capture	0	X
Error code capture	–	ECC	Error Code Capture	0	X
Error warning limit	–	EWLR	Error Warning Limit Register	96	X
RX error counter	–	RXERR	Receive Error Counter	0 (reset)	X; note 4
TX error counter	–	TXERR	Transmit Error Counter	0 (reset)	X; note 4
TX buffer	–	TXB	Transmit Buffer	X	X
RX buffer	–	RXB	Receive Buffer	X; note 5	X; note 5
ACR 0 to 3	–	ACR0 to ACR3	Acceptance Code Registers	X	X
AMR 0 to 3	–	AMR0 to AMR3	Acceptance Mask Registers	X	X
RX message counter	–	RMC	RX Message Counter	0	0
RX buffer start address	–	RBSA	RX Buffer Start Address	00000000	X
Clock divider	–	CDR	Clock Divider Register	00000000 Intel; 00000101 Motorola	X

Notes

1. X means that the value of these registers or bits is not influenced.
2. Remarks in brackets explain functional meaning.
3. On bus-off the error warning interrupt is set, if enabled.
4. If the reset mode was entered due to a bus-off condition, the receive error counter is cleared and the transmit error counter is initialized to 127 to count-down the CAN-defined bus-off recovery time consisting of 128 occurrences of 11 consecutive recessive bits.
5. Internal read/write pointers of the RXFIFO are reset to their initial values. A subsequent read access to the RXB would show undefined data values (parts of old messages).
   If a message is transmitted, this message is written in parallel to the receive buffer. A receive interrupt is generated only if this transmission was forced by the self reception request. So, even if the receive buffer is empty, the last transmitted message may be read from the receive buffer until it is overwritten by the next received or transmitted message.
   Upon a hardware reset, the RXFIFO pointers are reset to the physical RAM address '0'. Setting CR.0 by software or due to the bus-off event will reset the RXFIFO pointers to the currently valid FIFO start address (RBSA register) which is different from the RAM address '0' after the first release receive buffer command.

6.4.3 MODE REGISTER (MOD)

The contents of the mode register are used to change the behaviour of the CAN controller. Bits may be set or reset by the CPU which uses the control register as a read/write memory. Reserved bits are read as logic 0.

Table 12 Bit interpretation of the mode register (MOD); CAN address '0'

BIT	SYMBOL	NAME	VALUE	FUNCTION
MOD.7	–	–	–	reserved
MOD.6	–	–	–	reserved
MOD.5	–	–	–	reserved
MOD.4	SM	Sleep Mode; note 1	1	sleep; the CAN controller enters sleep mode if no CAN interrupt is pending and if there is no bus activity
			0	wake-up; the CAN controller wakes up if sleeping
MOD.3	AFM	Acceptance Filter Mode; note 2	1	single; the single acceptance filter option is enabled (one filter with the length of 32 bit is active)
			0	dual; the dual acceptance filter option is enabled (two filters, each with the length of 16 bit are active)
MOD.2	STM	Self Test Mode; note 2	1	self test; in this mode a full node test is possible without any other active node on the bus using the self reception request command; the CAN controller will perform a successful transmission, even if there is no acknowledge received
			0	normal; an acknowledge is required for successful transmission



BIT	SYMBOL	NAME	VALUE	FUNCTION
MOD.1	LOM	Listen Only Mode; notes 2 and 3	1	listen only; in this mode the CAN would give no acknowledge to the CAN-bus, even if a message is received successfully
			0	normal; the error counters are stopped at the current value
MOD.0	RM	Reset Mode; note 4	1	reset; detection of a set reset mode bit results in aborting the current transmission/reception of a message and entering the reset mode
			0	normal; on the '1-to-0' transition of the reset mode bit, the CAN controller returns to the operating mode

Notes

1. The SJA1000 will enter sleep mode if the sleep mode bit is set to logic 1 (sleep); then there is no bus activity and no interrupt is pending. Setting of SM with at least one of the previously mentioned exceptions valid will result in a wake-up interrupt. After sleep mode is set, the CLKOUT signal continues until at least 15 bit times have passed, to allow a host microcontroller clocked via this signal to enter its own standby mode before the CLKOUT goes LOW. The SJA1000 will wake up when one of the three previously mentioned conditions is negated: after SM is set LOW (wake-up), there is bus activity or $\overline{INT}$ is driven LOW (active). On wake-up, the oscillator is started and a wake-up interrupt is generated. A sleeping SJA1000 which wakes up due to bus activity will not be able to receive this message until it detects 11 consecutive recessive bits (bus-free sequence). It should be noted that setting of SM is not possible in reset mode. After clearing of reset mode, setting of SM is possible first, when bus-free is detected again.
2. A write access to the bits MOD.1 to MOD.3 is only possible, if the reset mode is entered previously.
3. This mode of operation forces the CAN controller to be error passive. Message transmission is not possible. The listen only mode can be used e.g. for software driven bit rate detection and 'hot plugging'.
4. During a hardware reset or when the bus status bit is set to logic 1 (bus-off), the reset mode bit is also set to logic 1 (present). If this bit is accessed by software, a value change will become visible and takes effect first with the next positive edge of the internal clock which operates at half of the external oscillator frequency. During an external reset the microcontroller cannot set the reset mode bit to logic 0 (absent). Therefore, after having set the reset mode bit to logic 1, the microcontroller must check this bit to ensure that the external reset pin is not being held HIGH. Changes of the reset request bit are synchronized with the internal divided clock. Reading the reset request bit reflects the synchronized status. After the reset mode bit is set to logic 0 the CAN controller will wait for:
   a) One occurrence of bus-free signal (11 recessive bits), if the preceding reset has been caused by a hardware reset or a CPU-initiated reset.
   b) 128 occurrences of bus-free, if the preceding reset has been caused by a CAN controller initiated bus-off, before re-entering the bus-on mode.

6.4.4 COMMAND REGISTER (CMR)

A command bit initiates an action within the transfer layer of the CAN controller. This register is write only, all bits will return a logic 0 when being read. Between two commands at least one internal clock cycle is needed in order to proceed. The internal clock is half of the external oscillator frequency.

Table 13 Bit interpretation of the command register (CMR); CAN address 1

BIT	SYMBOL	NAME	VALUE	FUNCTION
CMR.7	–	reserved	–	–
CMR.6	–	reserved	–	–
CMR.5	–	reserved	–	–
CMR.4	SRR	Self Reception Request; notes 1 and 2	1	present; a message shall be transmitted and received simultaneously
			0	– (absent)
CMR.3	CDO	Clear Data Overrun; note 3	1	clear; the data overrun status bit is cleared
			0	– (no action)
CMR.2	RRB	Release Receive Buffer; note 4	1	released; the receive buffer, representing the message memory space in the RXFIFO is released
			0	– (no action)
CMR.1	AT	Abort Transmission; notes 5 and 2	1	present; if not already in progress, a pending transmission request is cancelled
			0	– (absent)
CMR.0	TR	Transmission Request; notes 6 and 2	1	present; a message shall be transmitted
			0	– (absent)

Notes

1. Upon self reception request a message is transmitted and simultaneously received if the acceptance filter is set to the corresponding identifier. A receive and a transmit interrupt will indicate correct self reception (see also self test mode in mode register).
2. Setting the command bits CMR.0 and CMR.1 simultaneously results in sending the transmit message once. No re-transmission will be performed in the event of an error or arbitration lost (single-shot transmission).
   Setting the command bits CMR.4 and CMR.1 simultaneously results in sending the transmit message once using the self reception feature. No re-transmission will be performed in the event of an error or arbitration lost.
   Setting the command bits CMR.0, CMR.1 and CMR.4 simultaneously results in sending the transmit message once as described for CMR.0 and CMR.1.
   The moment the transmit status bit is set within the status register, the internal transmission request bit is cleared automatically.
   Setting CMR.0 and CMR.4 simultaneously will ignore the set CMR.4 bit.
3. This command bit is used to clear the data overrun condition indicated by the data overrun status bit. As long as the data overrun status bit is set no further data overrun interrupt is generated.
4. After reading the contents of the receive buffer, the CPU can release this memory space in the RXFIFO by setting the release receive buffer bit to logic 1. This may result in another message becoming immediately available within the receive buffer. If there is no other message available, the receive interrupt bit is reset.

5. The abort transmission bit is used when the CPU requires the suspension of the previously requested transmission, e.g. to transmit a more urgent message before. A transmission already in progress is not stopped. In order to see if the original message has been either transmitted successfully or aborted, the transmission complete status bit should be checked. This should be done after the transmit buffer status bit has been set to logic 1 or a transmit interrupt has been generated.
It should be noted that a transmit interrupt is generated even if the message was aborted because the transmit buffer status bit changes to 'released'.
6. If the transmission request was set to logic 1 in a previous command, it cannot be cancelled by setting the transmission request bit to logic 0. The requested transmission may be cancelled by setting the abort transmission bit to logic 1.

6.4.5 STATUS REGISTER (SR)

The content of the status register reflects the status of the CAN controller. The status register appears to the CPU as a read only memory.

Table 14 Bit interpretation of the status register (SR); CAN address 2

BIT	SYMBOL	NAME	VALUE	FUNCTION
SR.7	BS	Bus Status; note 1	1	bus-off; the CAN controller is not involved in bus activities
			0	bus-on; the CAN controller is involved in bus activities
SR.6	ES	Error Status; note 2	1	error; at least one of the error counters has reached or exceeded the CPU warning limit defined by the Error Warning Limit Register (EWLR)
			0	ok; both error counters are below the warning limit
SR.5	TS	Transmit Status; note 3	1	transmit; the CAN controller is transmitting a message
			0	idle
SR.4	RS	Receive Status; note 3	1	receive; the CAN controller is receiving a message
			0	idle
SR.3	TCS	Transmission Complete Status; note 4	1	complete; last requested transmission has been successfully completed
			0	incomplete; previously requested transmission is not yet completed
SR.2	TBS	Transmit Buffer Status; note 5	1	released; the CPU may write a message into the transmit buffer
			0	locked; the CPU cannot access the transmit buffer; a message is either waiting for transmission or is in the process of being transmitted
SR.1	DOS	Data Overrun Status; note 6	1	overrun; a message was lost because there was not enough space for that message in the RXFIFO
			0	absent; no data overrun has occurred since the last clear data overrun command was given

BIT	SYMBOL	NAME	VALUE	FUNCTION
SR.0	RBS	Receive Buffer Status; note 7	1	full; one or more complete messages are available in the RXFIFO
			0	empty; no message is available

Notes

1. When the transmit error counter exceeds the limit of 255, the bus status bit is set to logic 1 (bus-off), the CAN controller will set the reset mode bit to logic 1 (present) and an error warning interrupt is generated, if enabled. The transmit error counter is set to 127 and the receive error counter is cleared. It will stay in this mode until the CPU clears the reset mode bit. Once this is completed the CAN controller will wait the minimum protocol-defined time (128 occurrences of the bus-free signal) counting down the transmit error counter. After that the bus status bit is cleared (bus-on), the error status bit is set to logic 0 (ok), the error counters are reset and an error warning interrupt is generated, if enabled. Reading the TX error counter during this time gives information about the status of the bus-off recovery.
2. Errors detected during reception or transmission will effect the error counters according to the CAN 2.0B protocol specification. The error status bit is set when at least one of the error counters has reached or exceeded the CPU warning limit (EWLR). An error warning interrupt is generated, if enabled. The default value of EWLR after hardware reset is 96.
3. If both the receive status and the transmit status bits are logic 0 (idle) the CAN-bus is idle. If both bits are set the controller is waiting to become idle again. After a hardware reset 11 consecutive recessive bits have to be detected until the idle status is reached. After bus-off this will take 128 of 11 consecutive recessive bits.
4. The transmission complete status bit is set to logic 0 (incomplete) whenever the transmission request bit or the self reception request bit is set to logic 1. The transmission complete status bit will remain at logic 0 until a message is transmitted successfully.
5. If the CPU tries to write to the transmit buffer when the transmit buffer status bit is logic 0 (locked), the written byte will not be accepted and will be lost without this being indicated.
6. When a message that is to be received has passed the acceptance filter successfully, the CAN controller needs space in the RXFIFO to store the message descriptor and for each data byte which has been received. If there is not enough space to store the message, that message is dropped and the data overrun condition is indicated to the CPU at the moment this message becomes valid. If this message is not completed successfully (e.g. due to an error), no overrun condition is indicated.
7. After reading all messages within the RXFIFO and releasing their memory space with the command release receive buffer this bit is cleared.



6.4.6 INTERRUPT REGISTER (IR)

The interrupt register allows the identification of an interrupt source. When one or more bits of this register are set, a CAN interrupt will be indicated to the CPU. After this register is read by the CPU all bits are reset except for the receive interrupt bit.

The interrupt register appears to the CPU as a read only memory.

Table 15 Bit interpretation of the interrupt register (IR); CAN address 3

BIT	SYMBOL	NAME	VALUE	FUNCTION
IR.7	BEI	Bus Error Interrupt	1	set; this bit is set when the CAN controller detects an error on the CAN-bus and the BEIE bit is set within the interrupt enable register
			0	reset
IR.6	ALI	Arbitration Lost Interrupt	1	set; this bit is set when the CAN controller lost the arbitration and becomes a receiver and the ALIE bit is set within the interrupt enable register
			0	reset
IR.5	EPI	Error Passive Interrupt	1	set; this bit is set whenever the CAN controller has reached the error passive status (at least one error counter exceeds the protocol-defined level of 127) or if the CAN controller is in the error passive status and enters the error active status again and the EPIE bit is set within the interrupt enable register
			0	reset
IR.4	WUI	Wake-Up Interrupt; note 1	1	set; this bit is set when the CAN controller is sleeping and bus activity is detected and the WUIE bit is set within the interrupt enable register
			0	reset
IR.3	DOI	Data Overrun Interrupt	1	set; this bit is set on a '0-to-1' transition of the data overrun status bit and the DOIE bit is set within the interrupt enable register
			0	reset
IR.2	EI	Error Warning Interrupt	1	set; this bit is set on every change (set and clear) of either the error status or bus status bits and the EIE bit is set within the interrupt enable register
			0	reset
IR.1	TI	Transmit Interrupt	1	set; this bit is set whenever the transmit buffer status changes from '0-to-1' (released) and the TIE bit is set within the interrupt enable register
			0	reset
IR.0	RI	Receive Interrupt; note 2	1	set; this bit is set while the receive FIFO is not empty and the RIE bit is set within the interrupt enable register
			0	reset; no more message is available within the RXFIFO

Notes

1. A wake-up interrupt is also generated, if the CPU tries to set the sleep bit while the CAN controller is involved in bus activities or a CAN interrupt is pending.
2. The behaviour of this bit is equivalent to that of the receive buffer status bit with the exception, that RI depends on the corresponding interrupt enable bit (RIE). So the receive interrupt bit is not cleared upon a read access to the interrupt register. Giving the command 'release receive buffer' will clear RI temporarily. If there is another message available within the FIFO after the release command, RI is set again. Otherwise RI remains cleared.

6.4.7 INTERRUPT ENABLE REGISTER (IER)

The register allows to enable different types of interrupt sources which are indicated to the CPU.

The interrupt enable register appears to the CPU as a read/write memory.

Table 16 Bit interpretation of the interrupt enable register (IER); CAN address 4

BIT	SYMBOL	NAME	VALUE	FUNCTION
IER.7	BEIE	Bus Error Interrupt Enable	1	enabled; if an bus error has been detected, the CAN controller requests the respective interrupt
			0	disabled
IER.6	ALIE	Arbitration Lost Interrupt Enable	1	enabled; if the CAN controller has lost arbitration, the respective interrupt is requested
			0	disabled
IER.5	EPIE	Error Passive Interrupt Enable	1	enabled; if the error status of the CAN controller changes from error active to error passive or vice versa, the respective interrupt is requested
			0	disabled
IER.4	WUIE	Wake-Up Interrupt Enable	1	enabled; if the sleeping CAN controller wakes up, the respective interrupt is requested
			0	disabled
IER.3	DOIE	Data Overrun Interrupt Enable	1	enabled; if the data overrun status bit is set (see status register; Table 14), the CAN controller requests the respective interrupt
			0	disabled
IER.2	EIE	Error Warning Interrupt Enable	1	enabled; if the error or bus status change (see status register; Table 14), the CAN controller requests the respective interrupt
			0	disabled
IER.1	TIE	Transmit Interrupt Enable	1	enabled; when a message has been successfully transmitted or the transmit buffer is accessible again (e.g. after an abort transmission command), the CAN controller requests the respective interrupt
			0	disabled
IER.0	RIE	Receive Interrupt Enable; note 1	1	enabled; when the receive buffer status is 'full' the CAN controller requests the respective interrupt
			0	disabled

Note

1. The receive interrupt enable bit has direct influence to the receive interrupt bit and the external interrupt output $\overline{INT}$. If RIE is cleared, the external $\overline{INT}$ pin will become HIGH immediately, if there is no other interrupt pending.

6.4.8 ARBITRATION LOST CAPTURE REGISTER (ALC)

This register contains information about the bit position of losing arbitration. The arbitration lost capture register appears to the CPU as a read only memory. Reserved bits are read as logic 0.

Table 17 Bit interpretation of the arbitration lost capture register (ALC); CAN address 11

BIT	SYMBOL	NAME	VALUE	FUNCTION
ALC.7 to ALC.5	–	reserved	For value and function see Table 18	
ALC.4	BITNO4	bit number 4		
ALC.3	BITNO3	bit number 3		
ALC.2	BITNO2	bit number 2		
ALC.1	BITNO1	bit number 1		
ALC.0	BITNO0	bit number 0		

On arbitration lost, the corresponding arbitration lost interrupt is forced, if enabled. At the same time, the current bit position of the bit stream processor is captured into the arbitration lost capture register. The content within this register is fixed until the users software has read out its contents once. The capture mechanism is then activated again.

The corresponding interrupt flag located in the interrupt register is cleared during the read access to the interrupt register. A new arbitration lost interrupt is not possible until the arbitration lost capture register is read out once.

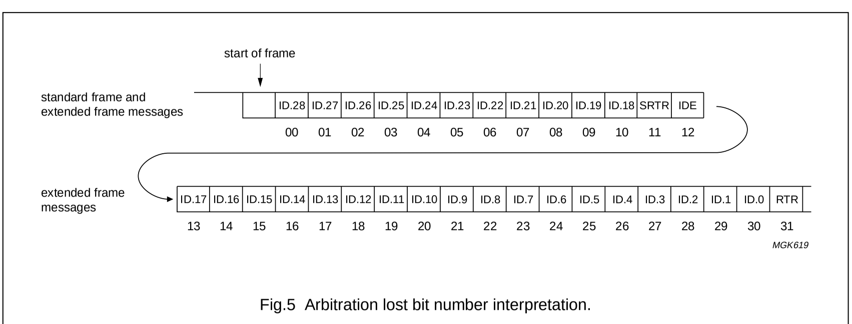


Fig.5 Arbitration lost bit number interpretation.

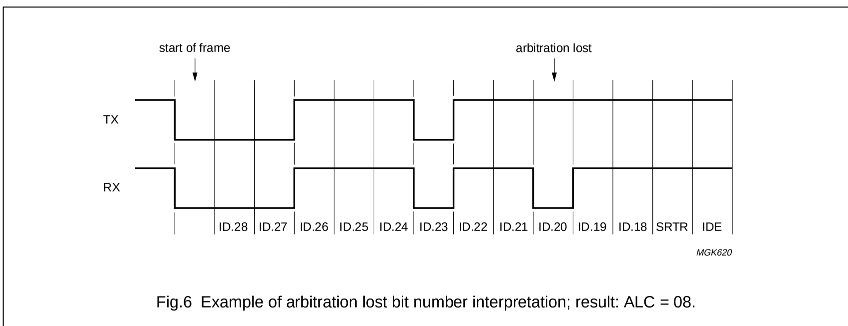


Fig.6 Example of arbitration lost bit number interpretation; result: ALC = 08.

Table 18 Function of bits 4 to 0 of the arbitration lost capture register

BITS[1]					DECIMAL VALUE	FUNCTION
ALC.4	ALC.3	ALC.2	ALC.1	ALC.0		
0	0	0	0	0	00	arbitration lost in bit 1 of identifier
0	0	0	0	1	01	arbitration lost in bit 2 of identifier
0	0	0	1	0	02	arbitration lost in bit 3 of identifier
0	0	0	1	1	03	arbitration lost in bit 4 of identifier
0	0	1	0	0	04	arbitration lost in bit 5 of identifier
0	0	1	0	1	05	arbitration lost in bit 6 of identifier
0	0	1	1	0	06	arbitration lost in bit 7 of identifier
0	0	1	1	1	07	arbitration lost in bit 8 of identifier
0	1	0	0	0	08	arbitration lost in bit 9 of identifier
0	1	0	0	1	09	arbitration lost in bit 10 of identifier
0	1	0	1	0	10	arbitration lost in bit 11 of identifier
0	1	0	1	1	11	arbitration lost in bit SRTR; note 2
0	1	1	0	0	12	arbitration lost in bit IDE
0	1	1	0	1	13	arbitration lost in bit 12 of identifier; note 3
0	1	1	1	0	14	arbitration lost in bit 13 of identifier; note 3
0	1	1	1	1	15	arbitration lost in bit 14 of identifier; note 3
1	0	0	0	0	16	arbitration lost in bit 15 of identifier; note 3
1	0	0	0	1	17	arbitration lost in bit 16 of identifier; note 3
1	0	0	1	0	18	arbitration lost in bit 17 of identifier; note 3
1	0	0	1	1	19	arbitration lost in bit 18 of identifier; note 3
1	0	1	0	0	20	arbitration lost in bit 19 of identifier; note 3
1	0	1	0	1	21	arbitration lost in bit 20 of identifier; note 3
1	0	1	1	0	22	arbitration lost in bit 21 of identifier; note 3
1	0	1	1	1	23	arbitration lost in bit 22 of identifier; note 3
1	1	0	0	0	24	arbitration lost in bit 23 of identifier; note 3
1	1	0	0	1	25	arbitration lost in bit 24 of identifier; note 3
1	1	0	1	0	26	arbitration lost in bit 25 of identifier; note 3
1	1	0	1	1	27	arbitration lost in bit 26 of identifier; note 3
1	1	1	0	0	28	arbitration lost in bit 27 of identifier; note 3
1	1	1	0	1	29	arbitration lost in bit 28 of identifier; note 3
1	1	1	1	0	30	arbitration lost in bit 29 of identifier; note 3
1	1	1	1	1	31	arbitration lost in bit RTR; note 3

Notes

1. Binary coded frame bit number where arbitration was lost.
2. Bit RTR for standard frame messages.
3. Extended frame messages only.

6.4.9 ERROR CODE CAPTURE REGISTER (ECC)

This register contains information about the type and location of errors on the bus. The error code capture register appears to the CPU as a read only memory.

Table 19 Bit interpretation of the error code capture register (ECC); CAN address 12

BIT	SYMBOL	NAME	VALUE	FUNCTION
ECC.7[1]	ERRC1	Error Code 1	–	–
ECC.6[1]	ERRC0	Error Code 0	–	–
ECC.5	DIR	Direction	1	RX; error occurred during reception
			0	TX; error occurred during transmission
ECC.4[2]	SEG4	Segment 4	–	–
ECC.3[2]	SEG3	Segment 3	–	–
ECC.2[2]	SEG2	Segment 2	–	–
ECC.1[2]	SEG1	Segment 1	–	–
ECC.0[2]	SEG0	Segment 0	–	–

Notes

1. For bit interpretation of bits ECC.7 and ECC.6 see Table 20.
2. For bit interpretation of bits ECC.4 to ECC.0 see Table 21.

Table 20 Bit interpretation of bits ECC.7 and ECC.6

BIT ECC.7	BIT ECC.6	FUNCTION
0	0	bit error
0	1	form error
1	0	stuff error
1	1	other type of error

Table 21 Bit interpretation of bits ECC.4 to ECC.0; note 1

BIT ECC.4	BIT ECC.3	BIT ECC.2	BIT ECC.1	BIT ECC.0	FUNCTION
0	0	0	1	1	start of frame
0	0	0	1	0	ID.28 to ID.21
0	0	1	1	0	ID.20 to ID.18
0	0	1	0	0	bit SRTR
0	0	1	0	1	bit IDE
0	0	1	1	1	ID.17 to ID.13
0	1	1	1	1	ID.12 to ID.5
0	1	1	1	0	ID.4 to ID.0
0	1	1	0	0	bit RTR
0	1	1	0	1	reserved bit 1
0	1	0	0	1	reserved bit 0
0	1	0	1	1	data length code
0	1	0	1	0	data field
0	1	0	0	0	CRC sequence
1	1	0	0	0	CRC delimiter
1	1	0	0	1	acknowledge slot
1	1	0	1	1	acknowledge delimiter
1	1	0	1	0	end of frame
1	0	0	1	0	intermission
1	0	0	0	1	active error flag
1	0	1	1	0	passive error flag
1	0	0	1	1	tolerate dominant bits
1	0	1	1	1	error delimiter
1	1	1	0	0	overload flag

Note

1. Bit settings reflect the current frame segment to distinguish between different error events.

If a bus error occurs, the corresponding bus error interrupt is always forced, if enabled. At the same time, the current position of the bit stream processor is captured into the error code capture register. The content within this register is fixed until the users software has read out its content once. The capture mechanism is then activated again.

The corresponding interrupt flag located in the interrupt register is cleared during the read access to the interrupt register. A new bus error interrupt is not possible until the capture register is read out once.

6.4.10 ERROR WARNING LIMIT REGISTER (EWLR)

The error warning limit can be defined within this register. The default value (after hardware reset) is 96. In reset mode this register appears to the CPU as a read/write memory. In operating mode it is read only.

Note, that a content change of the EWLR is only possible, if the reset mode was entered previously. An error status change (see status register; Table 14) and an error warning interrupt forced by the new register content will not occur until the reset mode is cancelled again.

Table 22 Bit interpretation of the error warning limit register (EWLR); CAN address 13

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
EWL.7	EWL.6	EWL.5	EWL.4	EWL.3	EWL.2	EWL.1	EWL.0

6.4.11 RX ERROR COUNTER REGISTER (RXERR)

The RX error counter register reflects the current value of the receive error counter. After a hardware reset this register is initialized to logic 0. In operating mode this register appears to the CPU as a read only memory. A write access to this register is possible only in reset mode.

If a bus-off event occurs, the RX error counter is initialized to logic 0. The time bus-off is valid, writing to this register has no effect.

Note, that a CPU-forced content change of the RX error counter is only possible, if the reset mode was entered previously. An error status change (see status register; Table 14), an error warning or an error passive interrupt forced by the new register content will not occur, until the reset mode is cancelled again.

Table 23 Bit interpretation of the RX error counter register (RXERR); CAN address 14

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RXERR.7	RXERR.6	RXERR.5	RXERR.4	RXERR.3	RXERR.2	RXERR.1	RXERR.0

6.4.12 TX ERROR COUNTER REGISTER (TXERR)

The TX error counter register reflects the current value of the transmit error counter.

In operating mode this register appears to the CPU as a read only memory. A write access to this register is possible only in reset mode. After a hardware reset this register is initialized to logic 0. If a bus-off event occurs, the TX error counter is initialized to 127 to count the minimum protocol-defined time (128 occurrences of the bus-free signal). Reading the TX error counter during this time gives information about the status of the bus-off recovery.

If bus-off is active, a write access to TXERR in the range from 0 to 254 clears the bus-off flag and the controller will wait for one occurrence of 11 consecutive recessive bits (bus-free) after the reset mode has been cleared.

Table 24 Bit interpretation of the TX error counter register (TXERR); CAN address 15

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TXERR.7	TXERR.6	TXERR.5	TXERR.4	TXERR.3	TXERR.2	TXERR.1	TXERR.0

Writing 255 to TXERR allows to initiate a CPU-driven bus-off event. It should be noted that a CPU-forced content change of the TX error counter is only possible, if the reset mode was entered previously. An error or bus status change (see status register; Table 14), an error warning or an error passive interrupt forced by the new register content will not occur until the reset mode is cancelled again. After leaving the reset mode, the new TX counter content is interpreted and the bus-off event is performed in the same way, as if it was forced by a bus error event. That means, that the reset mode is entered again, the TX error counter is initialized to 127, the RX counter is cleared and all concerned status and interrupt register bits are set.

Clearing of reset mode now will perform the protocol-defined bus-off recovery sequence (waiting for 128 occurrences of the bus-free signal).

If the reset mode is entered again before the end of bus-off recovery (TXERR > 0), bus-off keeps active and TXERR is frozen.



6.4.13 TRANSMIT BUFFER

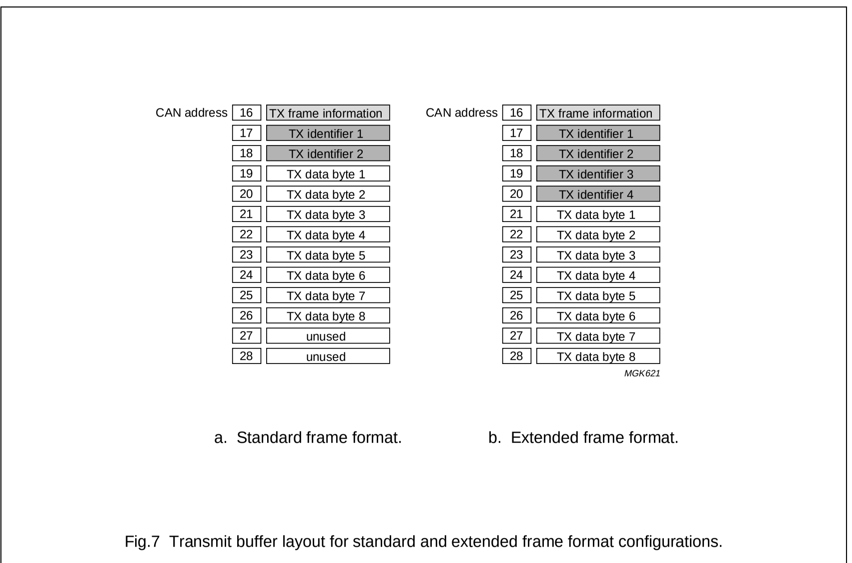
The global layout of the transmit buffer is shown in Fig.7. One has to distinguish between the Standard Frame Format (SFF) and the Extended Frame Format (EFF) configuration. The transmit buffer allows the definition of one transmit message with up to eight data bytes.

6.4.13.1 Transmit buffer layout

The transmit buffer layout is subdivided into descriptor and data fields where the first byte of the descriptor field is the frame information byte (frame information). It describes the frame format (SFF or EFF), remote or data frame and the data length. Two identifier bytes for SFF or four bytes for EFF messages follow. The data field contains up to eight data bytes.

The transmit buffer has a length of 13 bytes and is located in the CAN address range from 16 to 28.

Note, that a direct access to the transmit buffer RAM is possible using the CAN address space from 96 to 108. This RAM area is reserved for the transmit buffer. The three following bytes may be used for general purposes (CAN address 109, 110 and 111).



a. Standard frame format.   b. Extended frame format.

Fig.7 Transmit buffer layout for standard and extended frame format configurations.

6.4.13.2 Descriptor field of the transmit buffer

The bit layout of the transmit buffer is represented in Tables 25 to 27 for SFF and Tables 28 to 32 for EFF. The given configuration is chosen to be compatible with the receive buffer layout (see Section 6.4.14.1).

Table 25 TX frame information (SFF); CAN address 16

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
FF(1)	RTR(2)	X(3)	X(3)	DLC.3(4)	DLC.2(4)	DLC.1(4)	DLC.0(4)

Notes

1. Frame format.
2. Remote transmission request.
3. Don't care; recommended to be compatible to receive buffer (0) in case of using the self reception facility (self test).
4. Data length code bit.

Table 26 TX identifier 1 (SFF); CAN address 17; note 1

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ID.28	ID.27	ID.26	ID.25	ID.24	ID.23	ID.22	ID.21

Note

1. ID.X means identifier bit X.

Table 27 TX identifier 2 (SFF); CAN address 18; note 1

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ID.20	ID.19	ID.18	X(2)	X(3)	X(3)	X(3)	X(3)

Notes

1. ID.X means identifier bit X.
2. Don't care; recommended to be compatible to receive buffer (RTR) in case of using the self reception facility (self test).
3. Don't care; recommended to be compatible to receive buffer (0) in case of using the self reception facility (self test).

Table 28 TX frame information (EFF); CAN address 16

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
FF(1)	RTR(2)	X(3)	X(3)	DLC.3(4)	DLC.2(4)	DLC.1(4)	DLC.0(4)

Notes

1. Frame format.
2. Remote transmission request.
3. Don't care; recommended to be compatible to receive buffer (0) in case of using the self reception facility (self test).
4. Data length code bit.

Table 29 TX identifier 1 (EFF); CAN address 17; note 1

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ID.28	ID.27	ID.26	ID.25	ID.24	ID.23	ID.22	ID.21

Note

1. ID.X means identifier bit X.



Table 30 TX identifier 2 (EFF); CAN address 18; note 1

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ID.20	ID.19	ID.18	ID.17	ID.16	ID.15	ID.14	ID.13

Note

1. ID.X means identifier bit X.

Table 31 TX identifier 3 (EFF); CAN address 19; note 1

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ID.12	ID.11	ID.10	ID.9	ID.8	ID.7	ID.6	ID.5

Note

1. ID.X means identifier bit X.

Table 32 TX identifier 4 (EFF); CAN address 20; note 1

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ID.4	ID.3	ID.2	ID.1	ID.0	X(2)	X(3)	X(3)

Notes

1. ID.X means identifier bit X.
2. Don't care; recommended to be compatible to receive buffer (RTR) in case of using the self reception facility (self test).
3. Don't care; recommended to be compatible to receive buffer (0) in case of using the self reception facility (self test).

Table 33 Frame Format (FF) and Remote Transmission Request (RTR) bits

BIT	VALUE	FUNCTION
FF	1	EFF; extended frame format will be transmitted by the CAN controller
	0	SFF; standard frame format will be transmitted by the CAN controller
RTR	1	remote; remote frame will be transmitted by the CAN controller
	0	data; data frame will be transmitted by the CAN controller

6.4.13.3 Data Length Code (DLC)

The number of bytes in the data field of a message is coded by the data length code. At the start of a remote frame transmission the data length code is not considered due to the RTR bit being logic 1 (remote). This forces the number of transmitted/received data bytes to be 0. Nevertheless, the data length code must be specified correctly to avoid bus errors, if two CAN controllers start a remote frame transmission with the same identifier simultaneously.

The range of the data byte count is 0 to 8 bytes and is coded as follows:

DataByteCount = 8 × DLC.3 + 4 × DLC.2 + 2 × DLC.1 + DLC.0

For reasons of compatibility no data length code >8 should be used. If a value >8 is selected, 8 bytes are transmitted in the data frame with the Data Length Code specified in DLC.

6.4.13.4 Identifier (ID)

In Standard Frame Format (SFF) the identifier consists of 11 bits (ID.28 to ID.18) and in Extended Frame Format (EFF) messages the identifier consists of 29 bits (ID.28 to ID.0). ID.28 is the most significant bit, which is transmitted first on the bus during the arbitration process. The identifier acts as the message's name, used in a receiver for acceptance filtering, and also determines the bus access priority during the arbitration process.

The lower the binary value of the identifier the higher the priority. This is due to the larger number of leading dominant bits during arbitration.

6.4.13.5 Data field

The number of transferred data bytes is defined by the data length code. The first bit transmitted is the most significant bit of data byte 1 at CAN address 19 (SFF) or CAN address 21 (EFF).

6.4.14 RECEIVE BUFFER

The global layout of the receive buffer is very similar to the transmit buffer described in the previous section.
The receive buffer is the accessible part of the RXFIFO and is located in the range between CAN address 16 and 28. Each message is subdivided into a descriptor and a data field.

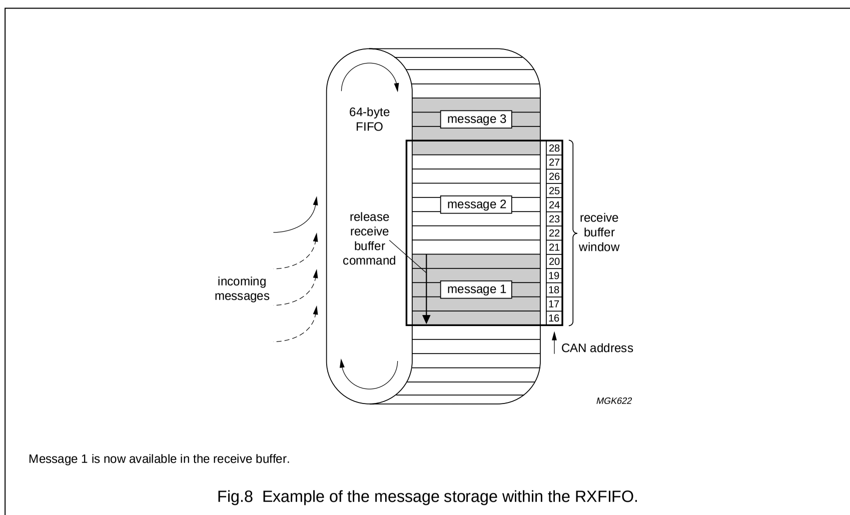


Fig.8 Example of the message storage within the RXFIFO.

6.4.14.1 Descriptor field of the receive buffer

The bit layout of the receive buffer is represented in Tables 34 to 36 for SFF and Tables 37 to 41 for EFF. The given configuration is chosen to be compatible with the transmit buffer layout (see Section 6.4.13.2).

Table 34 RX frame information (SFF); CAN address 16

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
FF(1)	RTR(2)	0	0	DLC.3(3)	DLC.2(3)	DLC.1(3)	DLC.0(3)

Notes

1. Frame format.
2. Remote transmission request.
3. Data length code bit.

Table 35 RX identifier 1 (SFF); CAN address 17; note 1

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ID.28	ID.27	ID.26	ID.25	ID.24	ID.23	ID.22	ID.21

Note

1. ID.X means identifier bit X.

Table 36 RX identifier 2 (SFF); CAN address 18; note 1

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ID.20	ID.19	ID.18	RTR(2)	0	0	0	0

Notes

1. ID.X means identifier bit X.
2. Remote transmission request.

Table 37 RX frame information (EFF); CAN address 16

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
FF(1)	RTR(2)	0	0	DLC.3(3)	DLC.2(3)	DLC.1(3)	DLC.0(3)

Notes

1. Frame format.
2. Remote transmission request.
3. Data length code bit.

Table 38 RX identifier 1 (EFF); CAN address 17; note 1

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ID.28	ID.27	ID.26	ID.25	ID.24	ID.23	ID.22	ID.21

Note

1. ID.X means identifier bit X.

Table 39 RX identifier 2 (EFF); CAN address 18; note 1

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ID.20	ID.19	ID.18	ID.17	ID.16	ID.15	ID.14	ID.13

Note

1. ID.X means identifier bit X.

Table 40 RX identifier 3 (EFF); CAN address 19; note 1

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ID.12	ID.11	ID.10	ID.9	ID.8	ID.7	ID.6	ID.5

Note

1. ID.X means identifier bit X.

Philips Semiconductors Preliminary specification

Stand-alone CAN controller SJA1000

Table 41 RX identifier 4 (EFF); can address 20; note 1

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ID.4	ID.3	ID.2	ID.1	ID.0	RTR(2)	0	0

Notes

1. ID.X means identifier bit X.
2. Remote transmission request.

Remark: the received data length code located in the frame information byte represents the real sent data length code, which may be greater than 8 (depends on sender). Nevertheless the maximum number of received data bytes is 8. This should be taken into account by reading a message from the receive buffer.

As described in Fig.8 the RXFIFO has space for 64 message bytes in total. It depends on the data length how many messages can fit in it at one time. If there is not enough space for a new message within the RXFIFO, the CAN controller generates a data overrun condition the moment this message becomes valid and the acceptance test was positive. A message which is partly written into the RXFIFO, when the data overrun situation occurs, is deleted. This situation is indicated to the CPU via the status register and the data overrun interrupt, if enabled.

6.4.15 ACCEPTANCE FILTER

With the help of the acceptance filter the CAN controller is able to allow passing of received messages to the RXFIFO only when the identifier bits of the received message are equal to the predefined ones within the acceptance filter registers.

The acceptance filter is defined by the Acceptance Code Registers (ACRn) and the Acceptance Mask Registers (AMRn). The bit patterns of messages to be received are defined within the acceptance code registers.
The corresponding acceptance mask registers allow to define certain bit positions to be 'don't care'.

Two different filter modes are selectable within the mode register (MOD.3, AFM; see Section 6.4.3):

- Single filter mode (bit AFM is logic 1)
- Dual filter mode (bit AFM is logic 0).

6.4.15.1 Single filter configuration

In this filter configuration one long filter (4-bytes) could be defined. The bit correspondences between the filter bytes and the message bytes depend on the currently received frame format.

Standard frame: if a standard frame format message is received, the complete identifier including the RTR bit and the first two data bytes are used for acceptance filtering. Messages may also be accepted if there are no data bytes existing due to a set RTR bit or if there is none or only one data byte because of the corresponding data length code.

For a successful reception of a message, all single bit comparisons have to signal acceptance.
Note, that the 4 least significant bits of AMR1 and ACR1 are not used. In order to be compatible with future products these bits should be programmed to be 'don't care' by setting AMR1.3, AMR1.2, AMR1.1 and AMR1.0 to logic 1.

Stand-alone CAN controller SJA1000

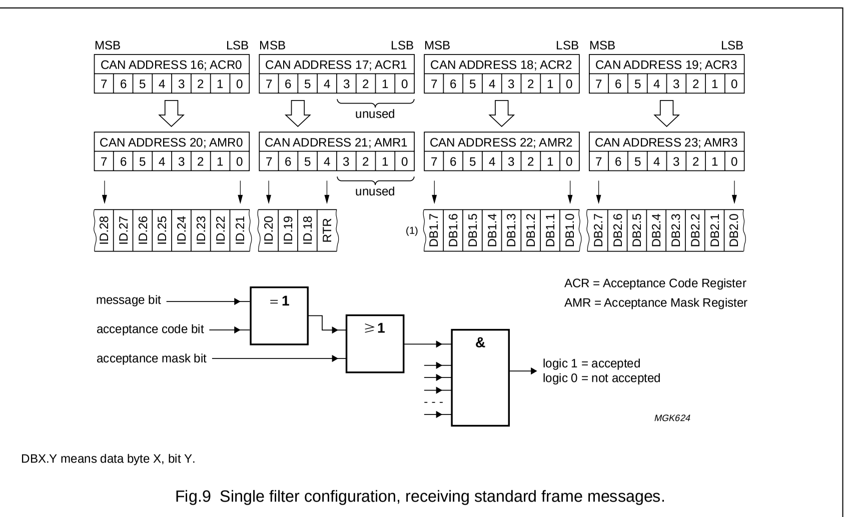


Fig.9 Single filter configuration, receiving standard frame messages.

Extended frame: if an extended frame format message is received, the complete identifier including the RTR bit is used for acceptance filtering.

For a successful reception of a message, all single bit comparisons have to signal acceptance.
It should be noted that the 2 least significant bits of AMR3 and ACR3 are not used. In order to be compatible with future products these bits should be programmed to be 'don't care' by setting AMR3.1 and AMR3.0 to logic 1.

Stand-alone CAN controller SJA1000

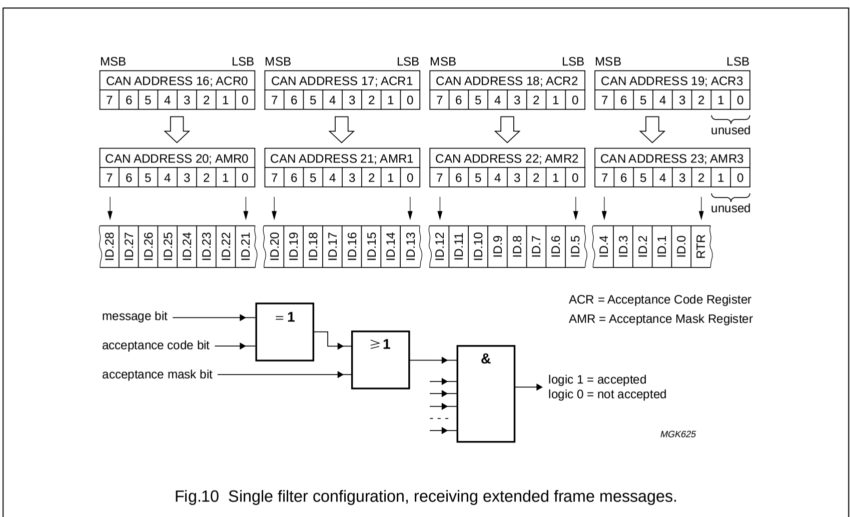


Fig.10 Single filter configuration, receiving extended frame messages.

6.4.15.2 *Dual filter configuration*

In this filter configuration two short filters can be defined. A received message is compared with both filters to decide, whether this message should be copied into the receive buffer or not. If at least one of the filters signals an acceptance, the received message becomes valid. The bit correspondences between the filter bytes and the message bytes depends on the currently received frame format.

Standard frame: if a standard frame message is received, the two defined filters are looking different. The first filter compares the complete standard identifier including the RTR bit and the first data byte of the message. The second filter just compares the complete standard identifier including the RTR bit.

For a successful reception of a message, all single bit comparisons of at least one complete filter have to signal acceptance. In case of a set RTR bit or a data length code of logic 0 no data byte is existing. Nevertheless a message may pass filter 1, if the first part up to the RTR bit signals acceptance.

If no data byte filtering is required for filter 1, the four least significant bits of AMR1 and AMR3 have to be set to logic 1 (don't care). Then both filters are working identically using the standard identifier range including the RTR bit.



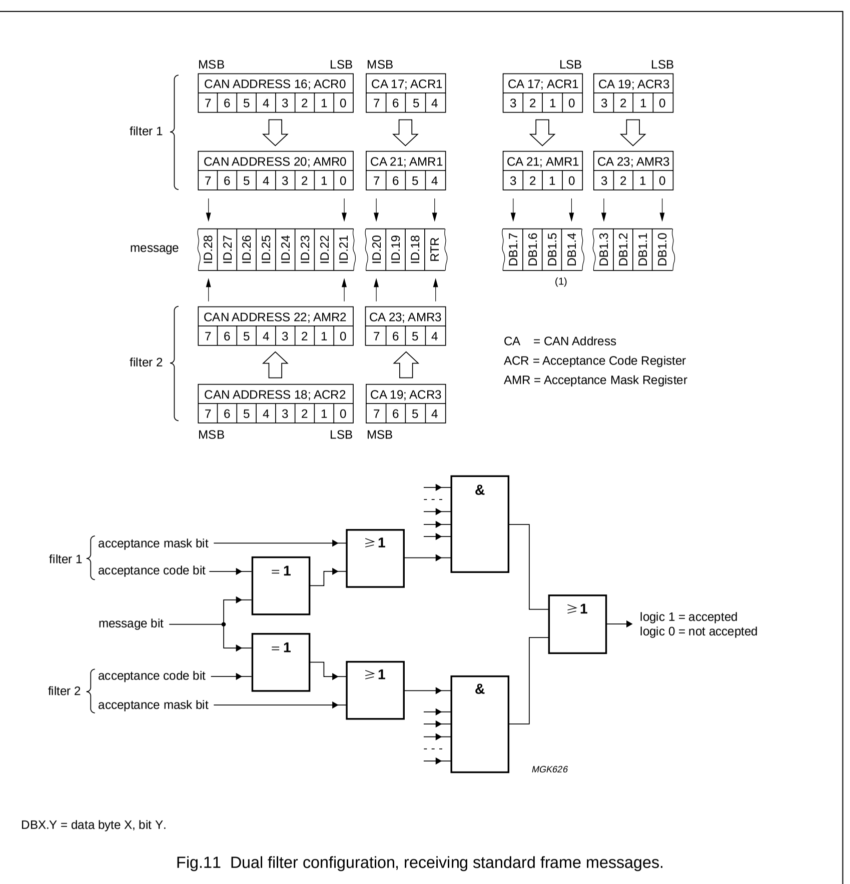


Fig.11 Dual filter configuration, receiving standard frame messages.

Extended frame: if an extended frame message is received, the two defined filters are looking identically. Both filters are comparing the first two bytes of the extended identifier range only.

For a successful reception of a message, all single bit comparisons of at least one complete filter have to indicate acceptance.

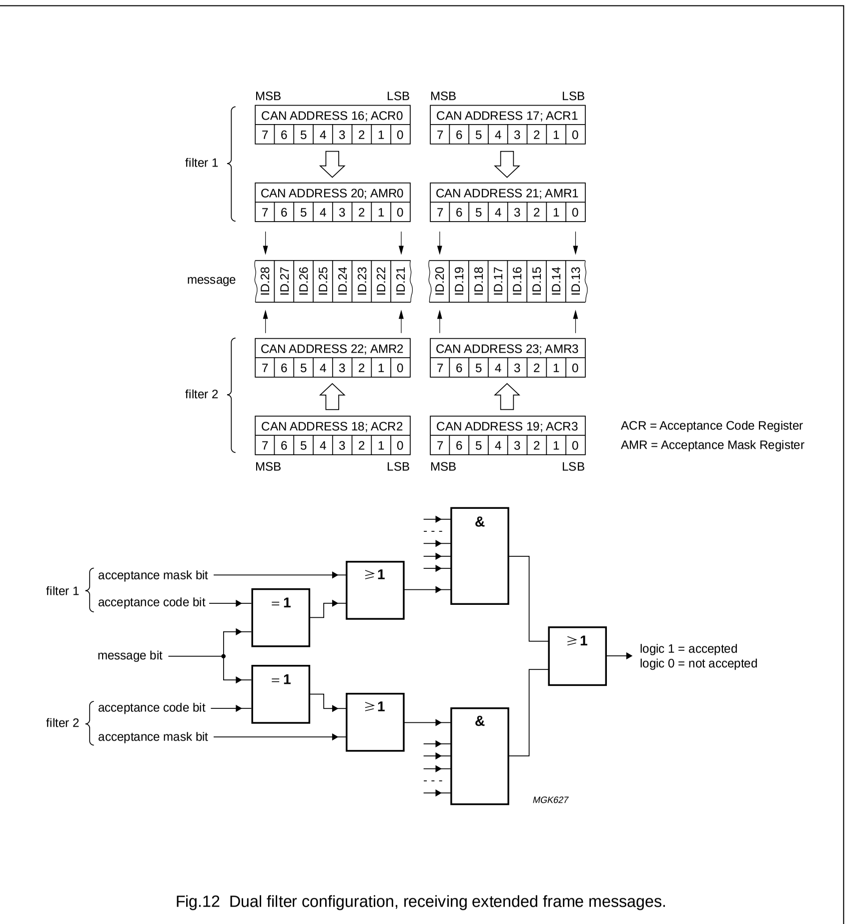


Fig.12 Dual filter configuration, receiving extended frame messages.

6.4.16 RX MESSAGE COUNTER (RMC)

The RMC register (CAN address 29) reflects the number of messages available within the RXFIFO. The value is incremented with each receive event and decremented by the release receive buffer command. After any reset event, this register is cleared.

Table 42 Bit interpretation of the RX message counter (RMC); CAN address 29

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
(0)(1)	(0)(1)	(0)(1)	RMC.4	RMC.3	RMC.2	RMC.1	RMC.0

Note

1. This bit cannot be written. During read-out of this register always a zero is given.

6.4.17 RX BUFFER START ADDRESS REGISTER (RBSA)

The RBSA register (CAN address 30) reflects the currently valid internal RAM address, where the first byte of the received message, which is mapped to the receive buffer window, is stored. With the help of this information it is possible to interpret the internal RAM contents.
The internal RAM address area begins at CAN address 32 and may be accessed by the CPU for reading and writing (writing in reset mode only).

Example: if RBSA is set to 24 (decimal), the current message visible in the receive buffer window
(CAN address 16 to 28) is stored within the internal RAM beginning at RAM address 24. Because the RAM is also mapped directly to the CAN address space beginning at CAN address 32 (equal to RAM address 0) this message may also be accessed using CAN address 56 and the following bytes
(CAN address = RBSA + 32 > 24 + 32 = 56).

If a message exceeds RAM address 63, it continues at RAM address 0.

The release receive buffer command is always given while there is at least one more message available within the FIFO. RBSA is updated to the beginning of the next message.

On hardware reset, this pointer is initialized to '00H'. Upon a software reset (setting of reset mode) this pointer keeps its old value, but the FIFO is cleared; this means that the RAM contents are not changed, but the next received (or transmitted) message will override the currently visible message within the receive buffer window.

The RX buffer start address register appears to the CPU as a read only memory in operating mode and as read/write memory in reset mode. It should be noted that a write access to RBSA takes effect first after the next positive edge of the internal clock frequency, which is half of the external oscillator frequency.

Table 43 Bit interpretation of the RX buffer start address register (RBSA); CAN address 30

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
(0)(1)	(0)(1)	RBSA.5	RBSA.4	RBSA.3	RBSA.2	RBSA.1	RBSA.0

Note

1. This bit cannot be written. During read-out of this register always a zero is given.

6.5 Common registers

6.5.1 BUS TIMING REGISTER 0 (BTR0)

The contents of the bus timing register 0 defines the values of the Baud Rate Prescaler (BRP) and the Synchronization Jump Width (SJW). This register can be accessed (read/write) if the reset mode is active.

In operating mode this register is read only, if the PeliCAN mode is selected. In BasicCAN mode a 'FFH' is reflected.

Table 44 Bit interpretation of bus timing register 0 (BTR0); CAN address 6

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SJW.1	SJW.0	BRP.5	BRP.4	BRP.3	BRP.2	BRP.1	BRP.0

6.5.1.1 Baud Rate Prescaler (BRP)

The period of the CAN system clock t_{scl} is programmable and determines the individual bit timing. The CAN system clock is calculated using the following equation:

$t_{scl} = 2 \times t_{CLK} \times (32 \times BRP.5 + 16 \times BRP.4 + 8 \times BRP.3 + 4 \times BRP.2 + 2 \times BRP.1 + BRP.0 + 1)$

where t_{CLK} = time period of the XTAL frequency = $\frac{1}{f_{XTAL}}$

6.5.1.2 Synchronization Jump Width (SJW)

To compensate for phase shifts between clock oscillators of different bus controllers, any bus controller must re-synchronize on any relevant signal edge of the current transmission. The synchronization jump width defines the maximum number of clock cycles a bit period may be shortened or lengthened by one re-synchronization:

$t_{SJW} = t_{scl} \times (2 \times SJW.1 + SJW.0 + 1)$

6.5.2 BUS TIMING REGISTER 1 (BTR1)

The contents of bus timing register 1 defines the length of the bit period, the location of the sample point and the number of samples to be taken at each sample point. This register can be accessed (read/write) if the reset mode is active.

In operating mode, this register is read only, if the PeliCAN mode is selected. In BasicCAN mode a 'FFH' is reflected.

Table 45 Bit interpretation of bus timing register 1 (BTR1); CAN address 7

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SAM	TSEG2.2	TSEG2.1	TSEG2.0	TSEG1.3	TSEG1.2	TSEG1.1	TSEG1.0

6.5.2.1 Sampling (SAM)

BIT	VALUE	FUNCTION
SAM	1	triple; the bus is sampled three times; recommended for low/medium speed buses (class A and B) where filtering spikes on the bus line is beneficial
	0	single; the bus is sampled once; recommended for high speed buses (SAE class C)



6.5.2.2 Time Segment 1 (TSEG1) and Time Segment 2 (TSEG2)

TSEG1 and TSEG2 determine the number of clock cycles per bit period and the location of the sample point, where:

$t_{SYNCSEG} = 1 \times t_{scl}$

$t_{TSEG1} = t_{scl} \times (8 \times TSEG1.3 + 4 \times TSEG1.2 + 2 \times TSEG1.1 + TSEG1.0 + 1)$

$t_{TSEG2} = t_{scl} \times (4 \times TSEG2.2 + 2 \times TSEG2.1 + TSEG2.0 + 1)$

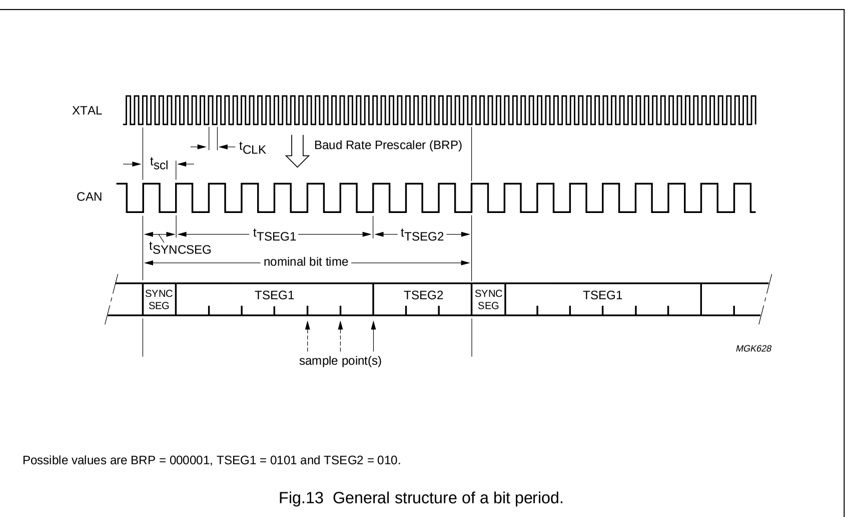


Fig.13 General structure of a bit period.

6.5.3 OUTPUT CONTROL REGISTER (OCR)

The output control register allows the set-up of different output driver configurations under software control.

This register may be accessed (read/write) if the reset mode is active. In operating mode, this register is read only, if the PeliCAN mode is selected. In BasicCAN mode a 'FFH' is reflected.

Table 46 Bit interpretation of the output control register (OCR); CAN address 8

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
OCTP1	OCTN1	OCPOL1	OCTP0	OCTN0	OCPOL0	OCMODE1	OCMODE0

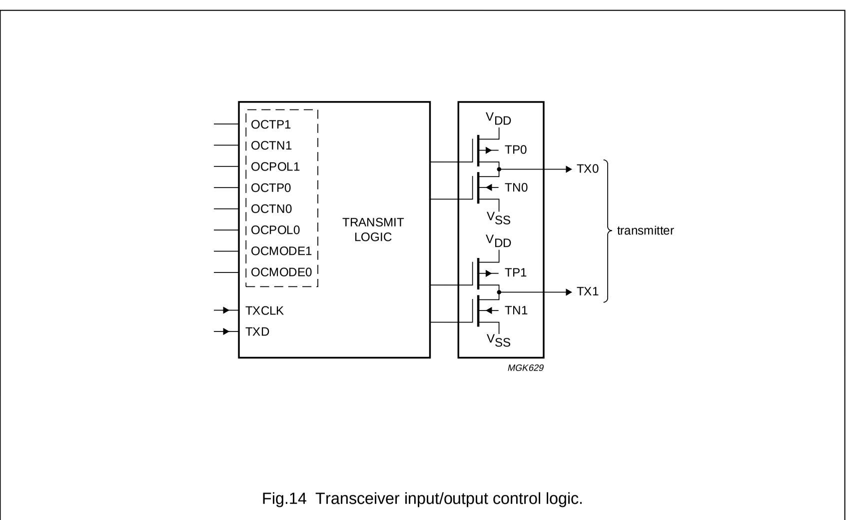


Fig.14 Transceiver input/output control logic.

If the SJA1000 is in the sleep mode a recessive level is output on the TX0 and TX1 pins with respect to the contents within the output control register. If the SJA1000 is in the reset state (reset request = HIGH) or the external reset pin $\overline{RST}$ is pulled LOW the outputs TX0 and TX1 are floating.

The transmit output stage is able to operate in different modes. Table 47 shows the output control register settings.

Table 47 Interpretation of OCMODE bits

OCMODE1	OCMODE0	DESCRIPTION
0	0	bi-phase output mode
0	1	test output mode; note 1
1	0	normal output mode
1	1	clock output mode

Note

1. In test output mode TXn will reflect the bit, detected on RX pins, with the next positive edge of the system clock. TN1, TN0, TP1 and TP0 are configured in accordance with the setting of OCR.

6.5.3.1 Normal output mode

In normal output mode the bit sequence (TXD) is sent via TX0 and TX1. The voltage levels on the output driver pins TX0 and TX1 depend on both the driver characteristic programmed by OCTPx, OCTNx (float, pull-up, pull-down, push-pull) and the output polarity programmed by OCPOLx.



6.5.3.2 *Clock output mode*

For the TX0 pin this is the same as in normal output mode. However, the data stream to TX1 is replaced by the transmit clock (TXCLK). The rising edge of the transmit clock (non-inverted) marks the beginning of a bit period. The clock pulse width is $1 \times t_{scl}$.

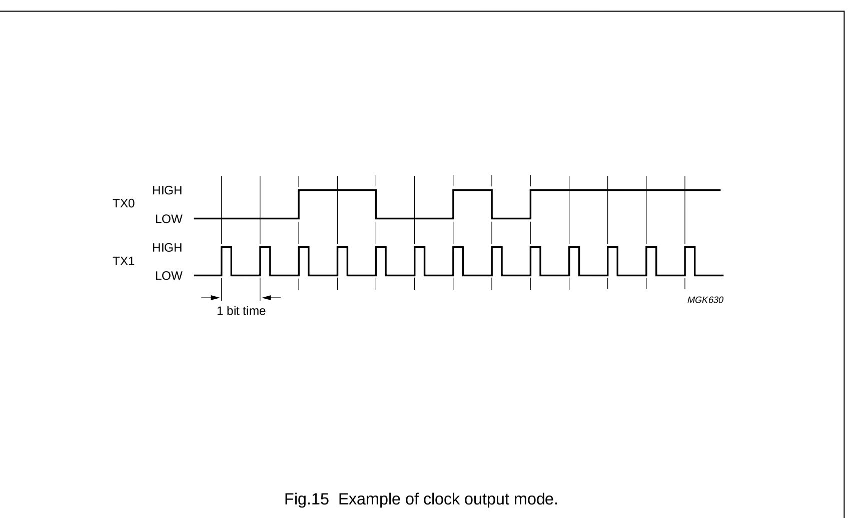


Fig.15 Example of clock output mode.

6.5.3.3 *Bi-phase output mode*

In contrast to the normal output mode the bit representation is time variant and toggled. If the bus controllers are galvanically decoupled from the bus line by a transformer, the bit stream is not allowed to contain a DC component. This is achieved by the following scheme.

During recessive bits all outputs are deactivated (floating). Dominant bits are sent with alternating levels on TX0 and TX1, i.e. the first dominant bit is sent on TX0, the second is sent on TX1, and the third one is sent on TX0 again, and so on. One possible configuration example of the bi-phase output mode timing is shown in Fig.16.

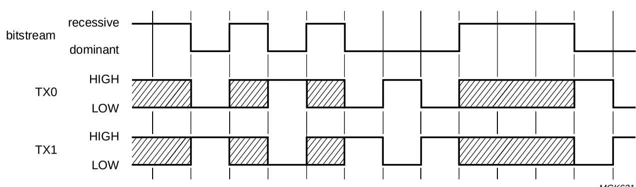


Fig.16 Bi-phase output mode example (output control register = F8H).

6.5.3.4 *Test output mode*

In test output mode the level connected to RX is reflected at TXn with the next positive edge of the system clock $\frac{f_{osc}}{2}$ corresponding to the programmed polarity in the output control register.

Table 48 shows the relationship between the bits of the output control register and the output pins TX0 and TX1.

Table 48 Output pin configuration; note 1

DRIVE	TXD	OCTPX	OCTNX	OCPOLX	TPX(2)	TNX(3)	TXX(4)
Float	X	0	0	X	off	off	float
Pull-down	0	0	1	0	off	on	LOW
	1	0	1	0	off	off	float
	0	0	1	1	off	off	float
	1	0	1	1	off	on	LOW
Pull-up	0	1	0	0	off	off	float
	1	1	0	0	on	off	HIGH
	0	1	0	1	on	off	HIGH
	1	1	0	1	off	off	float
Push-pull	0	1	1	0	off	on	LOW
	1	1	1	0	on	off	HIGH
	0	1	1	1	on	off	HIGH
	1	1	1	1	off	on	LOW

Notes

1. X = don't care.
2. TPX is the on-chip output transistor X, connected to V_{DD}.
3. TNX is the on-chip output transistor X, connected to V_{SS}.
4. TXX is the serial output level on pin TX0 or TX1. It is required that the output level on the CAN-bus line is dominant when TXD = 0 and recessive when TXD = 1.

The bit sequence (TXD) is sent via TX0 and TX1.
The voltage levels on the output driver pins depends on both the driver characteristics programmed by OCTP, OCTN (float, pull-up, pull-down, push-pull) and the output polarity programmed by OCPOL.

6.5.4 CLOCK DIVIDER REGISTER (CDR)

The clock divider register controls the CLKOUT frequency for the microcontroller and allows to deactivate the CLKOUT pin. Additionally a dedicated receive interrupt pulse on TX1, a receive comparator bypass and the selection between BasicCAN mode and PeliCAN mode is made here. The default state of the register after hardware reset is divide-by-12 for Motorola mode (00000101) and divide-by-2 for Intel mode (00000000).

On software reset (reset request/reset mode) this register is not influenced.

The reserved bit (CDR.4) will always reflect a logic 0. The application software should always write a logic 0 to this bit in order to be compatible with future features, which may be 1-active using this bit.

Table 49 Bit interpretation of the clock divider register (CDR); CAN address 31

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CAN mode	CBP	RXINTEN	(0)(1)	clock off	CD.2	CD.1	CD.0

Note

1. This bit cannot be written. During read-out of this register always a zero is given.

6.5.4.1 CD.2 to CD.0

The bits CD.2 to CD.0 are accessible without restrictions in reset mode as well as in operating mode. These bits are used to define the frequency at the external CLKOUT pin. For an overview of selectable frequencies see Table 50.

Table 50 CLKOUT frequency selection; note 1

CD.2	CD.1	CD.0	CLKOUT FREQUENCY
0	0	0	$\frac{f_{osc}}{2}$
0	0	1	$\frac{f_{osc}}{4}$
0	1	0	$\frac{f_{osc}}{6}$
0	1	1	$\frac{f_{osc}}{8}$
1	0	0	$\frac{f_{osc}}{10}$
1	0	1	$\frac{f_{osc}}{12}$
1	1	0	$\frac{f_{osc}}{14}$
1	1	1	f_{osc}

Note

1. f_{osc} is the frequency of the external oscillator (XTAL).

6.5.4.2 Clock off

Setting of this bit allows to disable the external CLKOUT pin of the SJA1000. A write access is possible only in reset mode (reset request bit is set in BasicCAN mode).

6.5.4.3 RXINTEN

This bit allows to use the TX1 output as a dedicated receive interrupt output. When a received message has passed the acceptance filter successfully, a receive interrupt pulse with the length of one bit time is always output at the TX1 pin (during the last bit of end of frame). The polarity and output drive are programmable via the output control register (see also Section 6.5.3). A write access is only possible in reset mode (the reset request bit is set in BasicCAN mode).

6.5.4.4 CBP

Setting of CDR.6 allows to bypass the CAN input comparator and is only possible in reset mode. This is useful in the event that the SJA1000 is connected to an external transceiver circuit. The internal delay of the SJA1000 is reduced, which will result in a longer maximum possible bus length. If CBP is set, only RX0 is active. The unused RX1 input should be connected to a defined level (e.g. V_{SS}).

6.5.4.5 CAN mode

CDR.7 defines the CAN mode. If CDR.7 is at logic 0 the CAN controller operates in BasicCAN mode. If set to logic 1 the CAN controller operates in PeliCAN mode. Write access is only possible in reset mode.



APPENDICE C: SCHEMI ELETTRICI

In questa appendice sono disponibili gli schemi elettrici delle interfaccie per la **GPC® 883** più frequentemente utilizzate. Tutte queste interfaccie possono essere prodotte autonomamente dall'utente mentre solo alcune di esse sono schede **grifo®** standard e possono quindi essere ordinate.

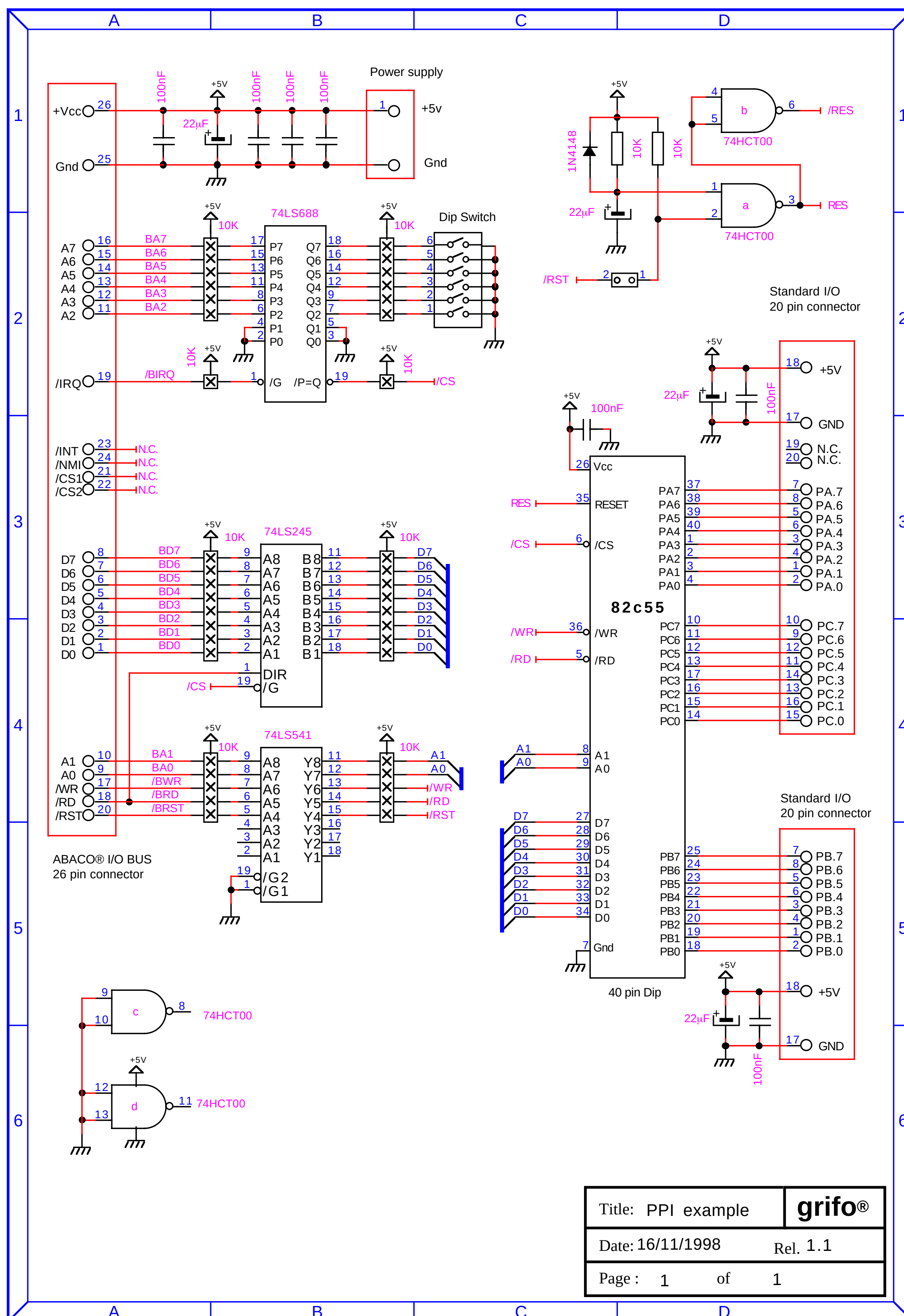


FIGURA C1: SCHEMA ELETTRICO DI ESPANSIONE PPI

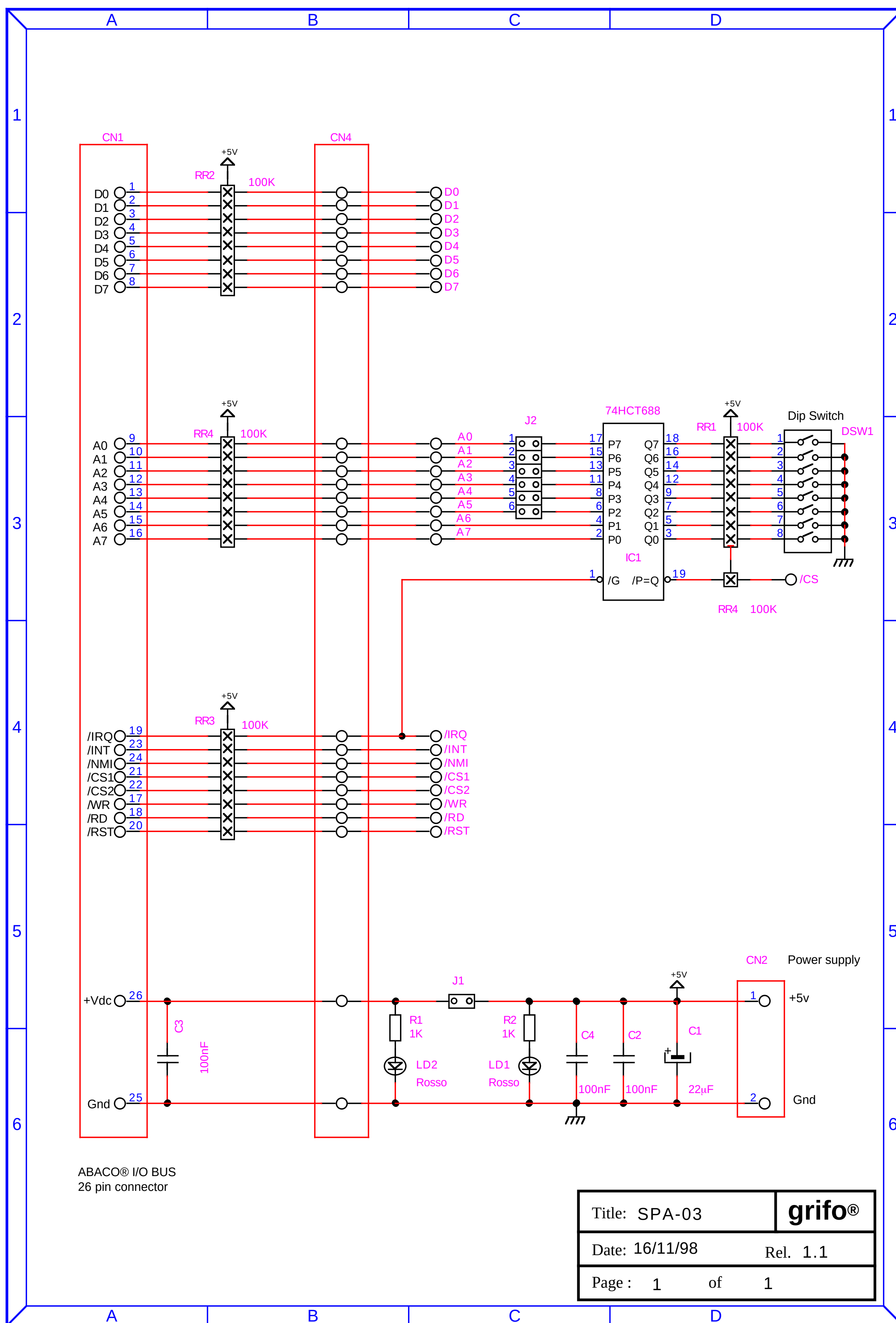


FIGURA C2: SCHEMA ELETTRICO SPA 03

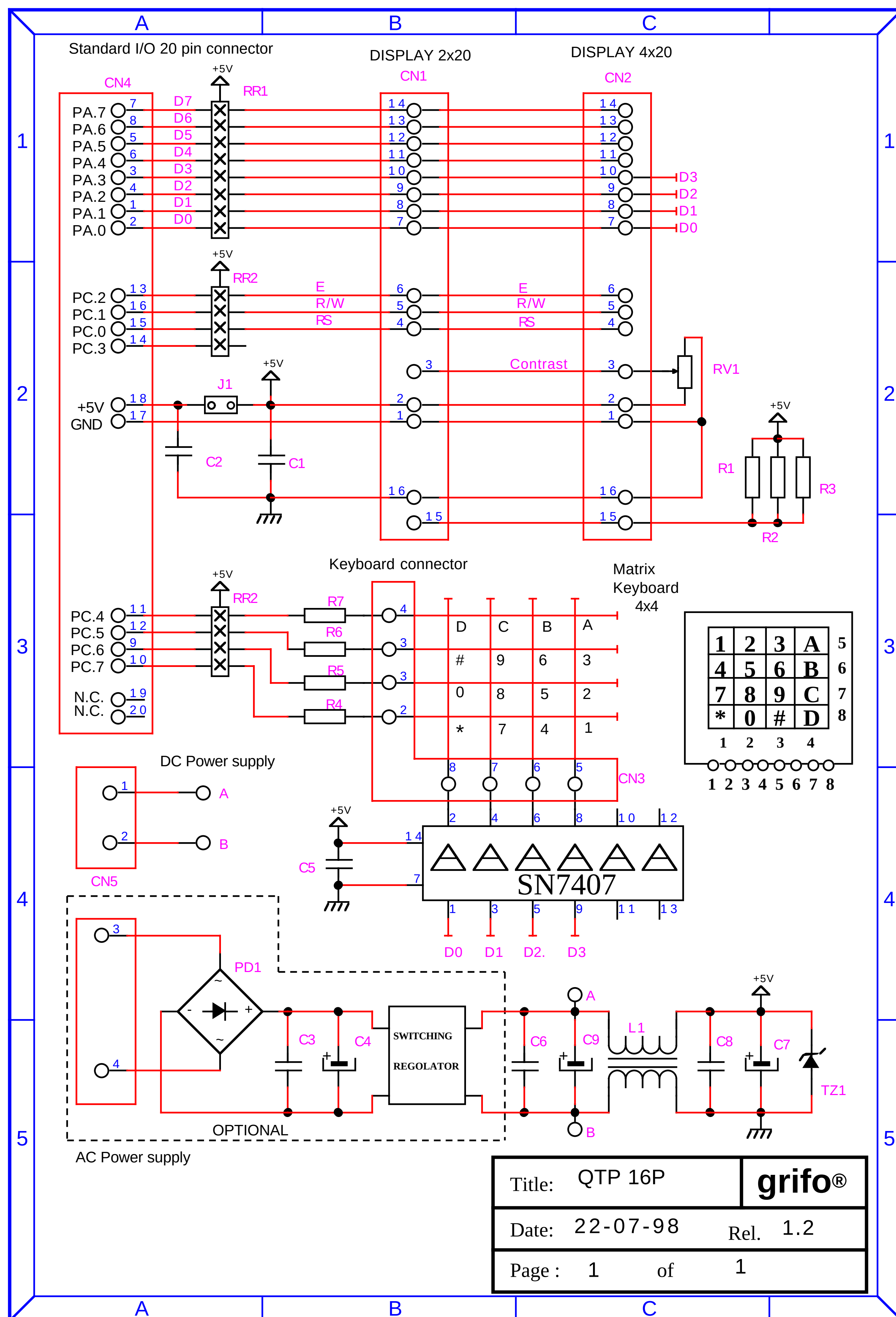


FIGURA C3: SCHEMA ELETTRICO QTP 16P

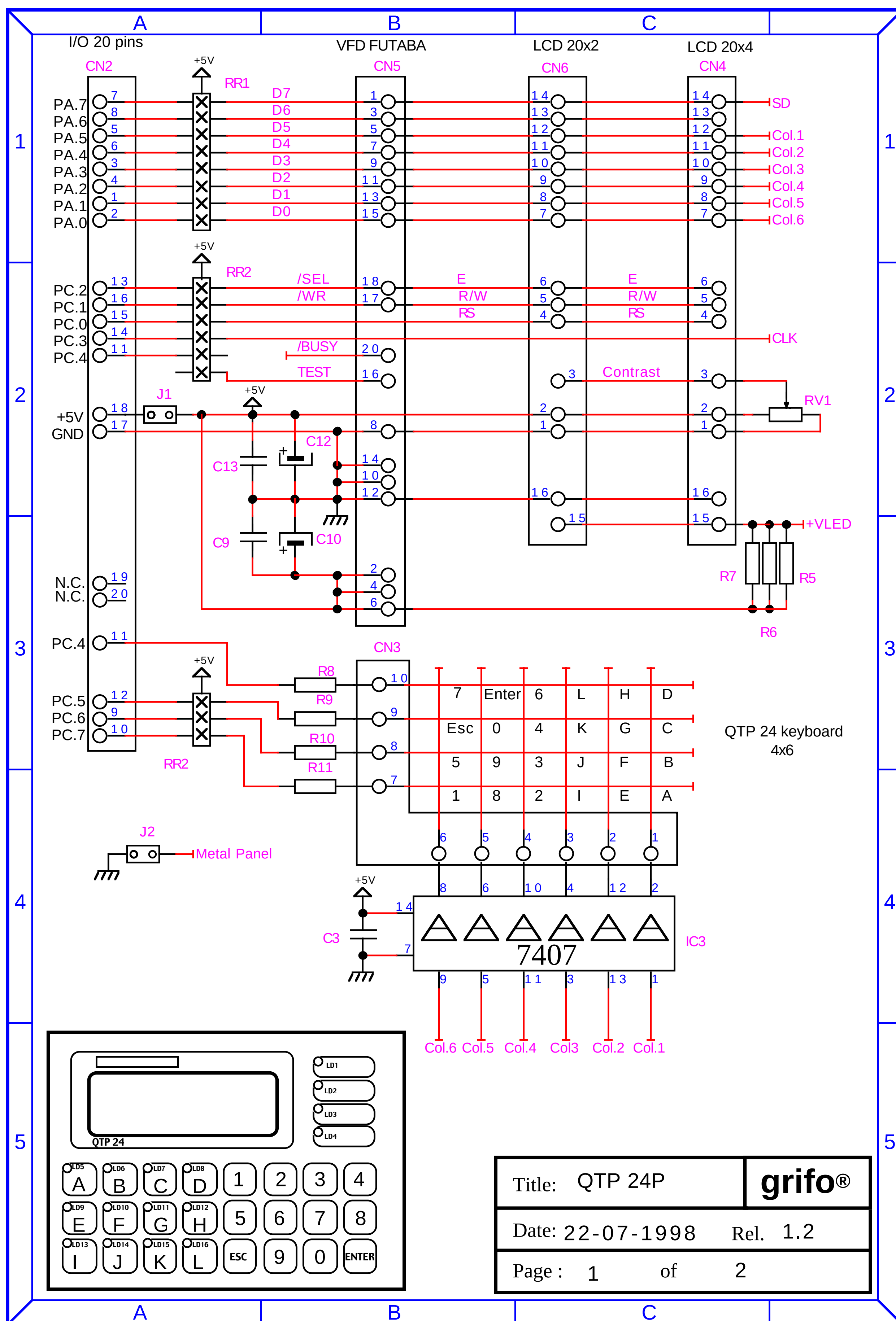


FIGURA C4: SCHEMA ELETTRICO QTP 24P (1 DI 2)

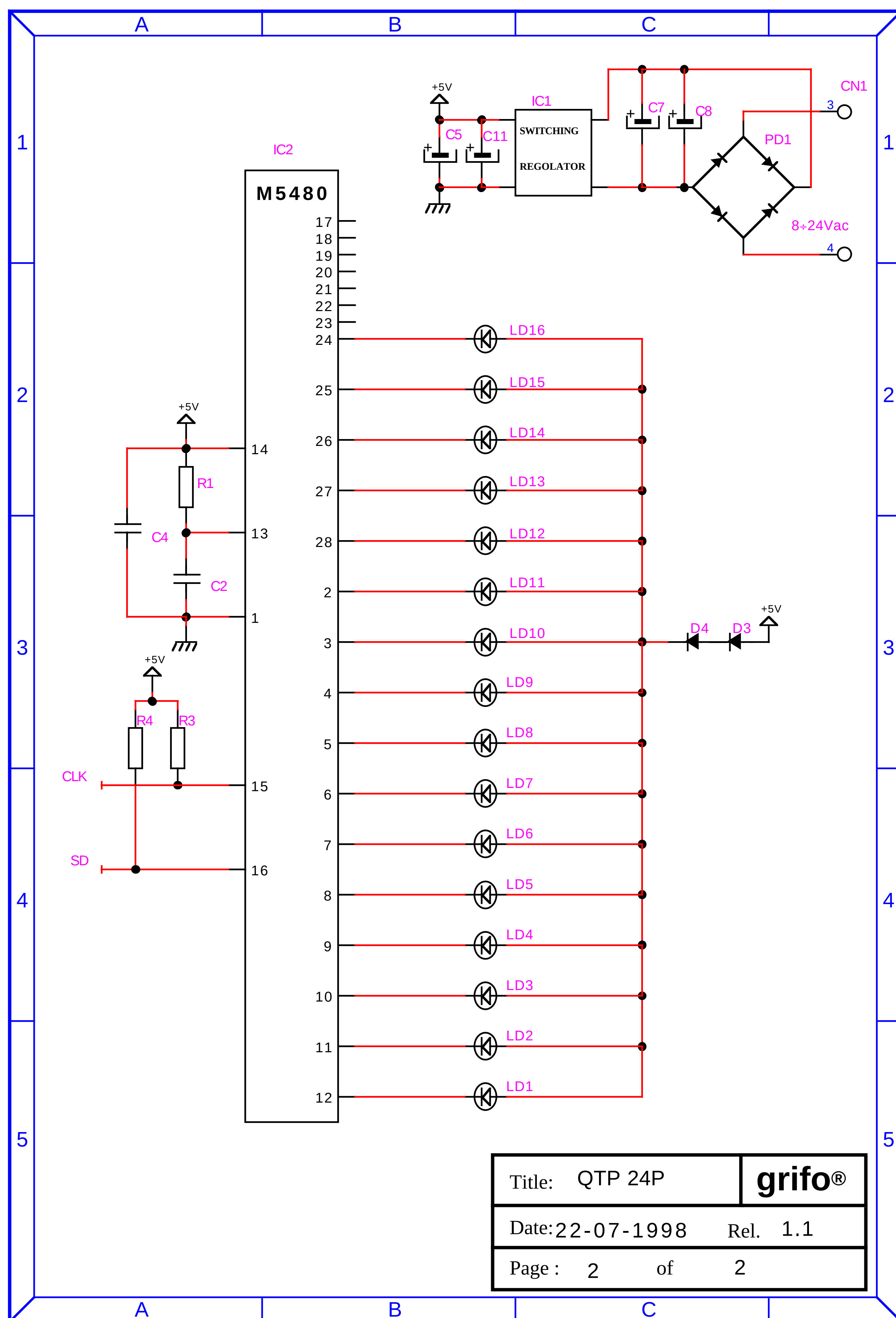


FIGURA C5: SCHEMA ELETTRICO QTP 24P (2 DI 2)

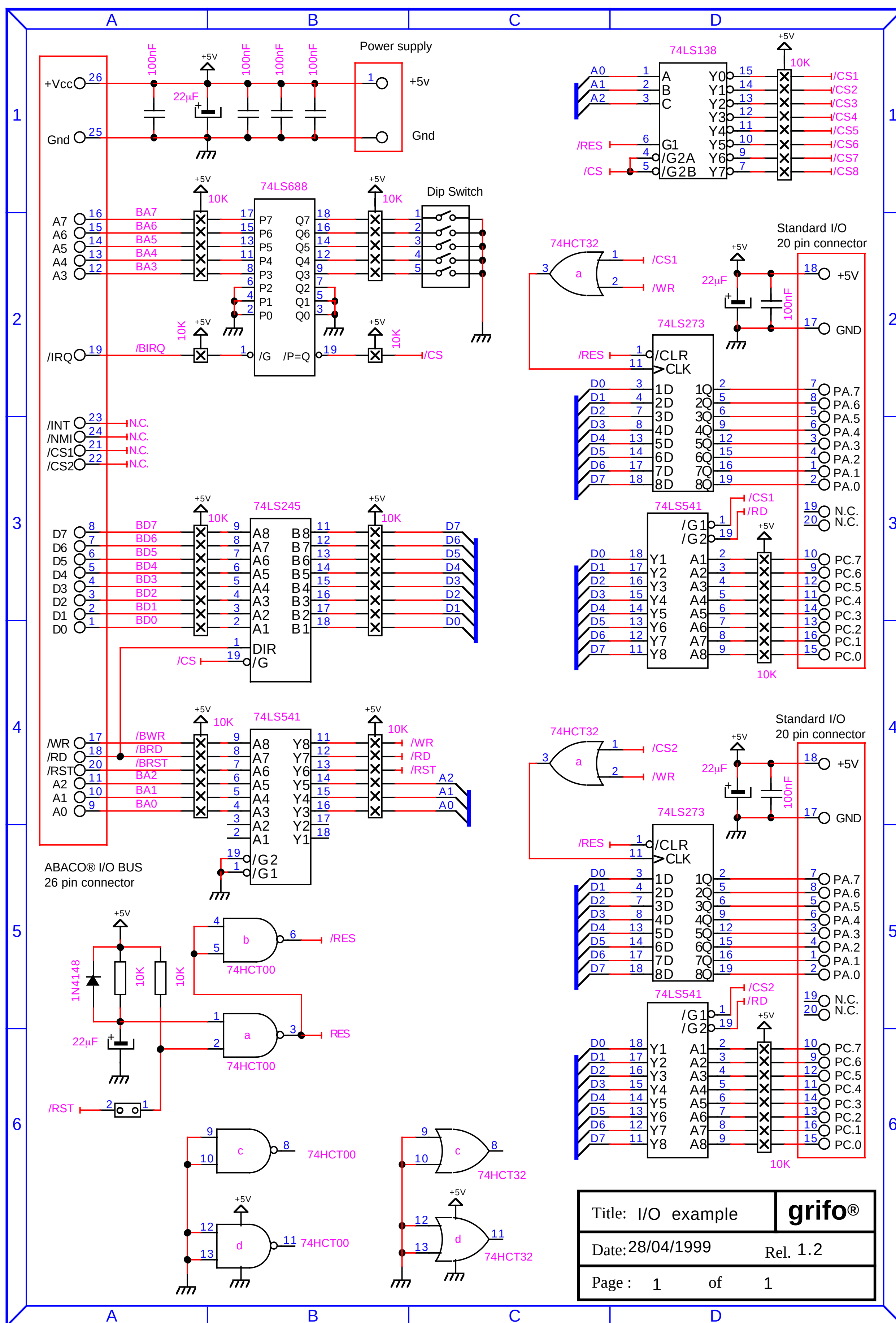


FIGURA C6: SCHEMA ELETTRICO DI INPUT OUTPUT SU ABACO® I/O BUS

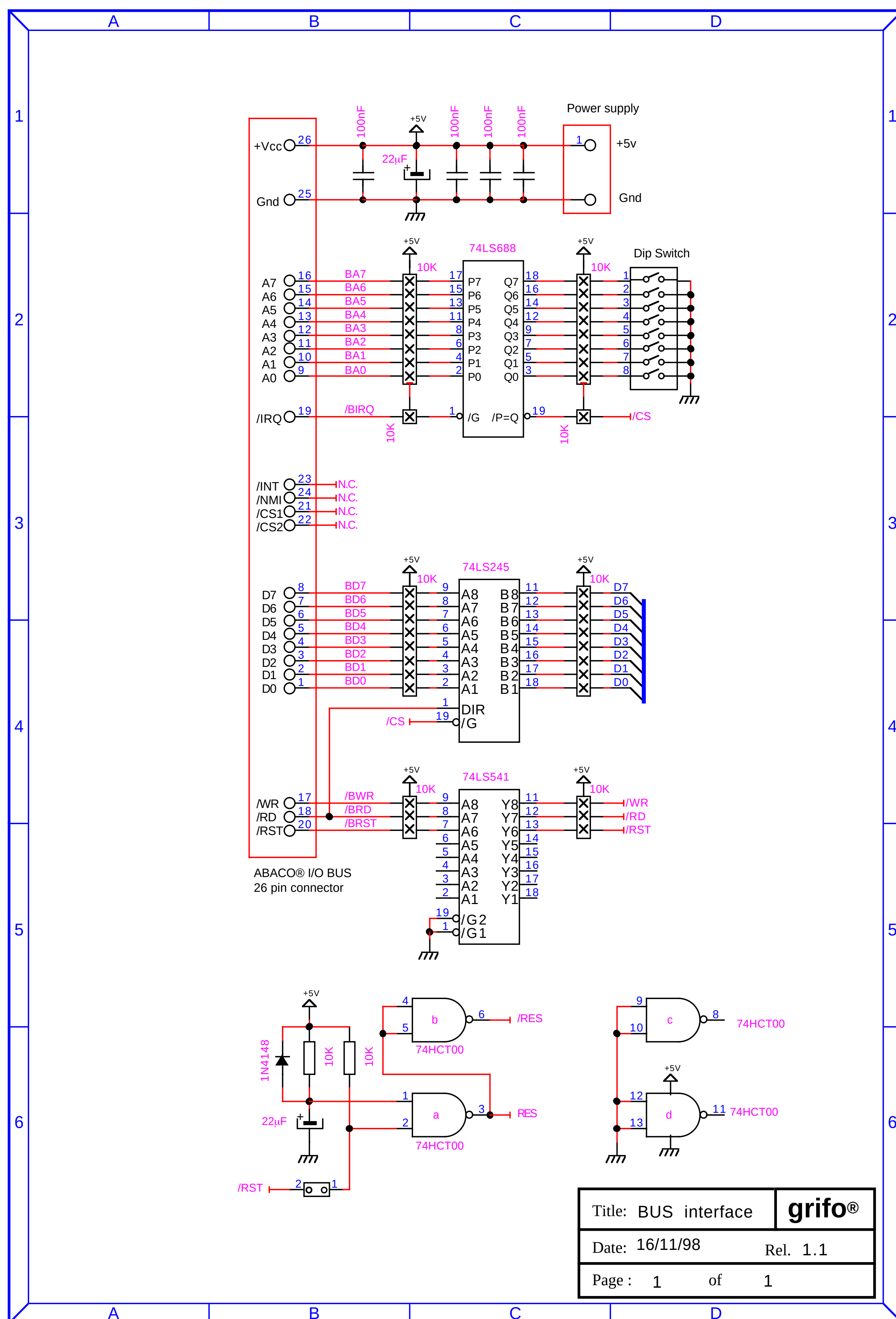


FIGURA C7: SCHEMA ELETTRICO INTERFACCIA BUS

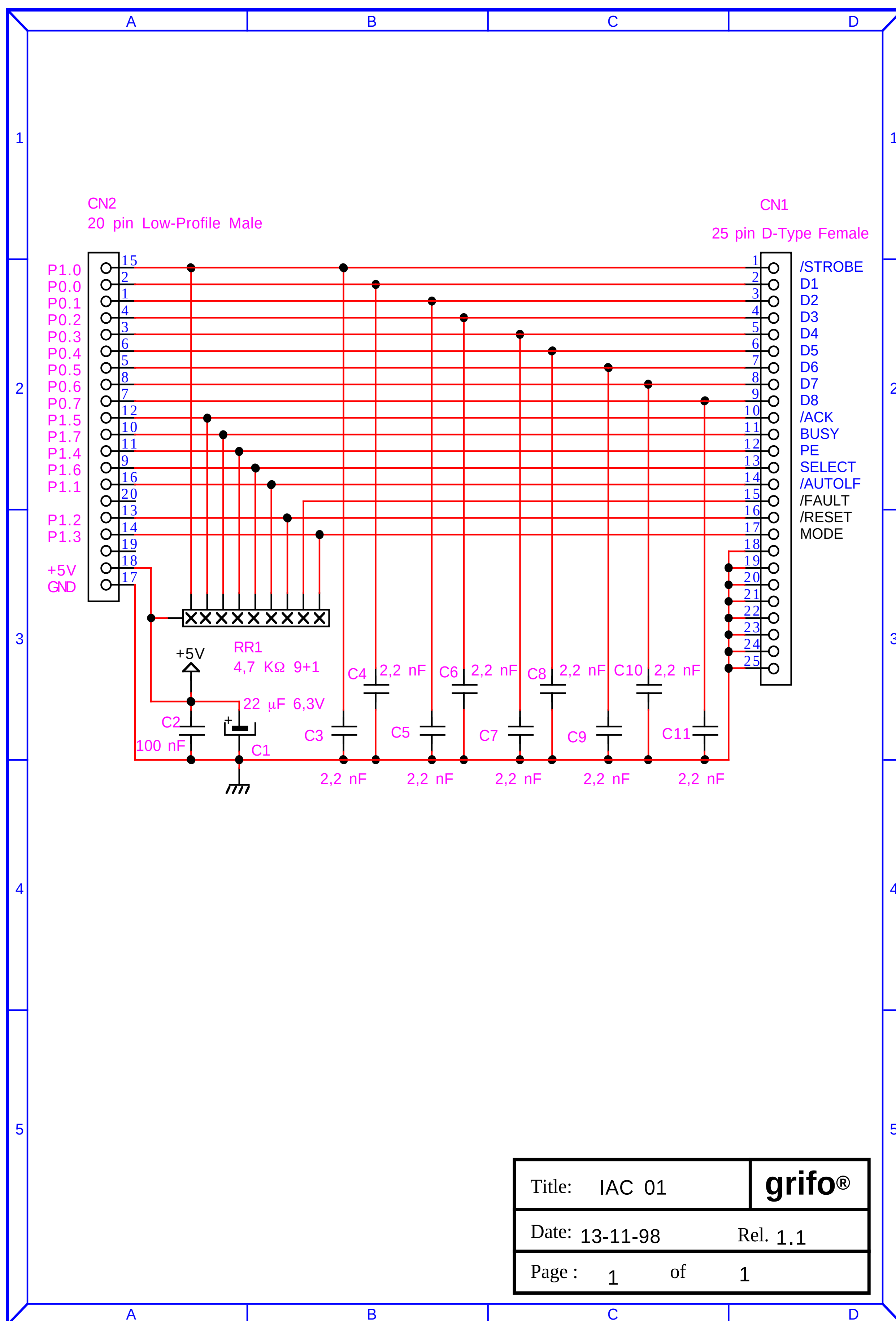


FIGURA C8: SCHEMA ELETTRICO IAC 01

APPENDICE D: INDICE ANALITICO

Simboli

A

B

C

L

M

O

P